Scenarios of copy-and-paste of mathematical objects

Paul Libbrecht, IU International University of Applied Science

Keywords: Copy-and-paste, Mathematical Documents, Web-Browsers, MathML, OpenMath, Authoring tools, Calculation Tools.

Abstract: Similarly to simple text, mathematical formulæ are a widely accepted language to represent ideas. Other representation of mathematical knowledge exist, e.g. in numeric or geometric forms. However, unlike text, mathematical objects are only transferrable (that is they can enter a transfer process) in limited and particular conditions, in part because encoding the formulæ is limited to encoding a graphical representation. An unsuccessful transfer leaves users with the task of transferring by redoing, retyping, or recreating the content.

Because automatic conversions may be working in many cases and because interoperable encoding and transfers could exist one day, this paper attempts to present scenarios of transfers of formulæ almost independently of what is doable currently. Inspired by the scenarios, a vision of interoperability could be built.

Introduction: The Context of Formula-Transfers

Mathematical knowledge is often represented using mathematical formulæ, a compact and often operation-expressive representation of the mathematical knowledge. The representation employs a vocabulary and grammar that, similarly to word-based language, varies in understandability depending on who is reading and writing. Mathematical formulæ may often be so well understoond that others (including systems) could operate on them in the very same fashion. Thus mathematical formulæ are an important ingredients of the generalization or knowledge transfer processes. Their slow development has been the vector of thoughts as can be seen, e.g. in [Mazur 2014]. We consider mathematical objects whose representations go slightly beyond mathematical notations: Any user-presentable mathematical object visualized through a computer document.

We are interested to the transfer of mathematical objects between documents: From documents presented to the user to documents created by the user who is using a form of authoring tool. Of particular importance is here the mathematical objects displayed in web-browsers as this family of application is the carryer of most of the knowledge exchanges of the world to date.

As a straight process, a transfer of mathematical object involves the user:

- seeing the content within the document
- marking as *selected* the parts of interest
- invoking a transfer function (such as *copy*, *cut*, *start-drag* or *share*)
- using an authoring tool and pointing to where one wants to introduce
- invoking the target of the transfer (e.g. the *paste* or *drop* operations)

The goal of this paper is to provide a sketch of selected scenarios of transfers between existing mathematical documents and authoring tools so as to question deeper where interoperability and where automation or development needs to happen for the scenarios to be doable.

This paper is the result of earlier research, e.g. [Libbrecht 2016], it takes in account the possible contribution of MathML 3 [MathML3] and OpenMath 2. It is motivated by the fact that, although the semantic web and visions such as the semantic clipboard [Berners-Lee 2004] exist since decades it should and could be possible to realize some of the scenarios below but very are actually doable thus far. Instead, partial automatisation or standards-efforts could make the scenarios be doable even if different standards are applied.

The core of this paper is made of a series of scenarios. Each is followed by comments so as to classify the scenarios. A few independent approaches to implementing them is articulated thereafter.

The scenarios

Transfer a formula to own notes

Our user is following a course including theorems and examples. She aims at assembling an own interpretation of the course as a way to learn, e.g. by gathering an e-portfolio. She often wants to copy sentences which she can, hypothetically, then paste adapt in her notes editing environment. They may include formulæ which should be copied along. This process of reformulation supports learning, e.g. when realizing a portfolio.

This transfer is very different from a simple storage of the files, pages of a book or a simple screenshot: Our user is touching up every bit so that the text becomes her own. That includes formulæ which the notes editing environment also offers her to edit.

- **Requirements**: Because formulæ are also copied along, they need to be representable in a way that is exchangeable between the web-page and the notes editing system. For example, the fragments could be transferred in HTML with MathML-presentation islands. Or the copied text could include a generic LaTeX encoding that would be suitable to paste within notes made in LaTeX.
- **Limits**: The selection should be working to identify the content being copied which may include parts of a formula. Moreover, page boundaries may give rise to unwished content when copying a selection that spans multiple pages.
- **Realisticness**: This scenario appears mostly not to work from HTML with MathML because (some of) MathML is generally wiped off of copied HTML.

Transfer a calculation to computing system

Our user used to know how to calculate volumes of tetrahedra but does not remember it anymore. She consults a web-page where the sketches describe her the necessary information which she can collect on her object of study.

The web-page contains formulæ which our user can select (only the right part of an equation) and on which she can invoke the copy command. She can paste into a computation environment such as a spreadsheet or computer algebra system. Because the formula is pasted, the user recognizes it by its graphical representation and by the name of the variables. She can adjust the variable names to match the names she had already input in her computing system.

The system can receive the formula and, provided it is given the values for each of the variables, is able to perform the calculation. Because of the automated calculation, our user is able to adjust the input and explore the interplay of the parameters.

- **Requirements**: computable formula (e.g. encoding in OpenMath or MathML-content) or automatically recognised formula: The computing system needs to receive the transfer and convert it to its internal representation; this internal representation should be user-editable as, otherwise, many formulæ will not be exactly relevant.
- **Limits**: Formulæ may involve functions not understandable by some computing systems: the paste function should support the user in transferring all but those missing "symbols" which the user might then resolve by alternative means.
- **Realisticness**: This process works in no spreadsheet that I could observe; it may work with computer algebra systems which can recognize the MathML or OpenMath source code after the user copies it in plain-text (through a special action).

Transfer a calculation to Wolfram|Alpha

Our user has been searching statistical tests to support her research in the data analysis. She found an online book that lists all possible tests and documents their requirements; one of them is particularly interesting but involves an integral which appears to have no discernable primitive. She remembers that Wolfram|Alpha might be usable to perform this but she needs to input the formula (which involves the Gamma function as well as fractions, roots and binomial coefficients, which she does not know how to input in Wolfram|Alpha. She takes her luck with a simple copy and paste from the integral.

Hypothetically, the web-page of Wolfram|Alpha has a mode to recognize the clipboard's mathematical nature and convert it to a Wolfram|Alpha input (e.g. in the Wolfram language). The result of this conversion is shown in plain-text and she presses return to see it rendered and to request its evaluation.

Because Wolfram|Alpha shows the intepretation, she can compare and recognise the conversion and adjust if need be. Moreover she is able to replace the variables with variables or numbers applicable to her data.

- **Requirements**: computable formula or automatic recognition thereof.
- **Limits**: as above, the convertibility of symbols is limited. Moreover, the ease of pasting may give rise to a sudden high demand which may require the infrastructure behind the mathematical search engine to be under high-demand.
- **Realisticness**: Some formulæ are simple enough that copying their plain text might make a copied text be actually wolfram|alpha-interpretable.

Transfer numbers into a spreadsheet

Our user uses a web-page to display sensor data and some graphics with them. However, she wants to make further analysis with the numbers and thus uses the reporting function which shows the various sensors' information in tables in web-pages. Our user is able to select the sensors display, using *rectangular selection* (a feature of Firefox) within the table to only extract the interesting numbers.

The numbers are annotated with units such as °C, kg, €, or m and are displayed using "." thousands separator and "," decimal separator.

Our user is able to paste to a spreadsheet so as to calculate with the numbers. Hypothetically, her paste recognizes the decimal separator, the thousands separator as well as the units. By using formulæ on the numbers just input, she is able to verify that the numbers are properly recognized and can be operated on.

- **Requirements**: The numbers need either to be semantically annotated to contain a standard representation of the numbers and the units (e.g. OpenMath or MathML-Content) or they are automatically reconized by the spreadsheet.
- **Limits**: The verification of the success of the conversion by calculating is not a complete proof of the validity of the conversion and, when users employ such a feature to transfer lots of data, it is easy to oversee that some conversion (or some sensors' input) is broken and needs to replaced or fixed before doing calculations (e.g. a broken sensor indicating an air-temperature of -999,99°C).
- **Realisticness**: While copying tables from a web-page to a spreadsheet works surprisingly well, the numbers conversion tends to only work if no units are present and the decimal and thousands separator are common with those of the web-page, a sad fact for the multilingual nature of the world wide web and of such data representation.

Transfer a construction to a geometry system

Our user is sketching a logo and remembers classical ornaments from past centuries which he would like to get inspiration from. He finds web-pages which describe the geometry rationales behind the ornaments and would like to report the geometry relationship to her logo sketches. While she can attempt to redo what is described in the book, she sees that several of the interesting ornaments are made with dynamic geometry so that one can influence the shape through the degrees of freedom of the construction. She decides to copy the "wireframe" structure of the constructions, not including the pictures of the ornaments, having properly selected its geometric objects before.

In her dynamic geometry environment she can then paste the construction while keeping the degrees of freedom. She can then apply the construction to the elementary symbols she decided to include in her logo.

- **Requirements**: Web-pages with dynamic geometry software presentations which allow selection and copy. Medium to carry the geometric constructions (e.g. a common file format). Dynamic-geometry system accepting the paste (all hypothetical).
- **Limits**: Incomplete convertibility of geometric constructions, e.g. because the exchange format is not rich enough, will long stay an issue and users will have to evaluate for themselves how much useful this can become.
- **Realisticness**: While the OpenMath format for dynamic geometry has been started, its implementations are lagging behind.

Transfer a formula to a LaTeX-based input

Our user is the writing of a complex mathematical works and does not remember how piecewise defined functions are defined. He thus searches the internet for "latex piecewise functions" and she finds documentation that explain how to do just this. The examples she finds provide examples that she needs to adjust for her function.

Because the first example she finds is working out of the box, she progressively modifies the contents so that the structure is kept but the elements are converted to the function she wants to define. Mutliple LaTeX-“compilation” give her confidence in mastering the use of the `\cases` which she did not know before.

- **Requirements**: The notation needs to be properly named and its name be used by others so that documentation pages exist on this topic.
- **Limits**: It is not rare to meet fragments of LaTeX that require extra packages (and sometimes explain what is needed to get the packages). Compatibility guarantees are rare and these solutions need to be avoided for environments that do not support macros (e.g. MathJax).
- **Realisticness**: This scenario can work in many cases.

Transfer a calculation code to a computation environment

Our user is now a data-scientist who needs to apply statistical and machine-learning calculations to analyse a data-set. She has the intuition that a process read in a conference article should provide an interesting step in her analysis. Because the paper exposes the code employed, she is able to copy it, hypothetically, from the paper and paste it in her programming environment, a Jupyter Notebook using the same programming language and libraries. However, she needs the code to be adjusted because her naming of the data-variables is different and she expects to see stronger effects, thus the paste is followed by multiple attempts of adaptation to make it run appropriately. She can test the appropriateness of the run by the lack of error but also by testing expected values derived from the data which she knows should occur.

- **Requirements**: Compatible and readable code as well as code that is expressed in a copyable fashion.
- **Limits**: Programmes that are big or that use libraries which our user does not know are difficult to trust or adapt. The capacities of the users need to match.
- **Realisticness**: This scenario can work in many cases. However, LaTeX-based PDF-files are often swallowing some whitespaces of code-fragments, which destroys, e.g., the structure of Python programmes. Similarly, hand-typed programme fragments in MS Word will replace quotes with litterary versions of the quotes which make them inappropriate for code pasting. Finally, in some rare cases, copied code contains invisible characters which may break at compilation.

Steps getting closer to the scenarios

In this section we describe a few performed or ongoing works that make the scenarios become more feasible.

Standardised encodings for mathematical objects have been studied since decades, this includes [MathML3], both in its presentation form and in its more computation-oriented format, more or less equivalent to [OpenMath2]. MathML3 names the clipboard flavours for MathML content and presentation which could, if implemented by the browsers, support strongly several scenarios. Moreover, HTML5 includes MathML3 elements which implies that transfers of HTML could include MathML.

Automatic Recognition appears in several scenarios to support or replace the interoperability between applications. It could permit an application receiving a transfer to recover the mathematical information behind a transferred object. For example, it may convert a graphically-encoded, LaTeX-based or mathml-presentation formula to a computable form, or it may recognize the units and decimal numbers representations of simple measurements. However such automatic recognitions tend to be highly sensitive to the culture and application-field of the user’s usage.

Unfortunately, the automatic conversion is often too complex to be parametrizable or modifiable, as, for example, indicated M.Panic when asked what the process was to enrich the Windows pen-based equation input [Panic 2009].

Selection so as to mark the content to be copied and the insertion point is often very specific to the editing environment while web-browsers mostly consider MathML-selection as text selection. A detail-structured MathML-expression (e.g. including enclosing the content of an integral within an `mrow`) has a better chance to let browsers select effectively.

Conclusion

We have presented a few scenarios that appear as natural expectations from users and which only rarely work in their full potential. Because copy and paste is the result of working by multiple implementors, it is only rarely guaranteed to work unless tested. An example where it appears to be tested is to copy from MS Word's equations to Wolfram|Alpha: The plain-text copied fragment can be close to a function that can be computed. It could be that more bilateral testing, in an automatic or manual fashion, is the only promising avenue so as to raise the percentage of successful formula transfers.

Literature

- Berners-Lee, T., 2004, Semantic Clipboard, W3C Design-Issues, available on the web.
- Buswell, S., Capprotti, O., Carlisle, D.P., Dewar, M.C., Gaëtano, M., Kolhase, M., Davenport, J.H., Ion, P., OpenMath2. Available from http://openmath.org/standard/om20-2017-07-22/omstd20-diff.html
- Carisle, D., Ion, P., Miner, R., MathML3, available from https://www.w3.org/TR/MathML/
- Mazur, B., 2014: Enlightening Symbols, Princeton University Press, see publisher page
- Libbrecht, P., Lokar, M., 2016, The plain text rap when copying mathematical formulæ, MathUI 2016.
- Panic, Marko, 2009, Math Handwriting Recognition in Windows 7 and Its Benefits, CICM 2009: Intelligent Computer Mathematics pp 29-30.