

## Executable HTML

***Citation for published version (APA):***

Batalas, N., Khan, V. J., & Markopoulos, P. (2021). Executable HTML. *SoftwareX*, 14, Article 100691.  
<https://doi.org/10.1016/j.softx.2021.100691>

***Document license:***

CC BY

***DOI:***

[10.1016/j.softx.2021.100691](https://doi.org/10.1016/j.softx.2021.100691)

***Document status and date:***

Published: 01/06/2021

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.



## Original Software Publication

## Executable HTML

Nikolaos Batalas\*, Vassilis-Javed Khan, Panos Markopoulos

Eindhoven University of Technology, Den Dolech 2, 5600MB, Eindhoven, The Netherlands



## ARTICLE INFO

## Article history:

Received 27 February 2020

Accepted 1 April 2021

## Keywords:

Experience sampling method

Ambulatory assessment

Ecological momentary assessment

Custom HTML elements

Executable HTML

## ABSTRACT

Executable HTML is a collection of custom HTML elements that offer document authors the ability to write and execute procedural logic in HTML5. We discuss the motivation of this approach, which lies in the composition of formal Ambulatory Assessment protocols for clinical psychology research, and provide an overview of the software's inner workings. For researchers in that field, the proposition allows a single HTML document to function as a both human-readable and computer-executable representation of a data collection protocol that can make easier the dissemination and replication of research.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Code metadata

Current code version	v0.0.1
Permanent link to code/repository used for this code version	<a href="https://github.com/ElsevierSoftwareX/SOFTX_2020_59">https://github.com/ElsevierSoftwareX/SOFTX_2020_59</a>
Code ocean compute capsule	None
Legal code license	BSD-3-Clause
Code versioning system used	git
Software code languages, tools, and services used	HTML, CSS, JavaScript
Compilation requirements, operating environments & dependencies	Browsers supporting Web components
Support email for questions	<a href="mailto:nikolaos.batalas@gmail.com">nikolaos.batalas@gmail.com</a>

## 1. Motivation and significance

## 1.1. Replication of ambulatory assessment research is hard

Clinical psychology employs a class of research methods known as Ambulatory Assessment (AA) methods, which aim to capture bio-psycho-social processes in the context of daily life [1]. These methods are nowadays mostly conducted via mobile devices such as smartphones and related peripherals. Respondents are sampled, sometimes multiple times per day, both for self-reports as responses to questionnaires (e.g. on feelings and thoughts), as well as for data streams from sensors on their devices (e.g., GPS locations, light intensity readings, count of steps taken, heart rate measurements).

In order to perform the data collection, researchers use software applications wherein they prepare the materials of the

study (e.g., questions) and specify the manner in which these materials are to be presented to respondents, (e.g. branching the questionnaire based on given responses, or setting times at which responses should be given). The combination of the materials used for the collection of data, and the manner in which they are served to participant, which in the case of AA studies consist largely of procedural logic, form the protocol of the study [2].

An important problem facing AA studies, is that research protocols are hard to publish in detail and re-implement [3,4]. In publications, researchers describe their protocols in natural language. Transcribing digital protocols into natural language in detail, and reconstructing them in software systems from natural language sources can be laborious and error-prone tasks. When describing computer programs, natural language can lead to ambiguities [5]. This hurts the replicability of AA research.

## 1.2. A singular representation on a universal platform can help

The Web-browser is a ubiquitous platform with rich functionality. It provides fertile ground for the construction of Web

\* Corresponding author.

E-mail address: [nikolaos.batalas@gmail.com](mailto:nikolaos.batalas@gmail.com) (Nikolaos Batalas).

**Table 1**  
Custom elements for procedural execution.

Element name	Description
aa-session	Root container for content to be injected, and manager of the data collection session
aa-sequence	Instantiates its children one at a time, as members of an ordered sequence. Does so also in response to received signals for elements that have blocking behavior (e.g., waiting for a user to press a button).
aa-choose	Container of conditional statements.
aa-when	Instantiates its contents when its condition is met.
aa-otherwise	Instantiates its contents when no conditions have been met.
aa-variable	Sets a variable value explicitly.
aa-jump	Tells the sequence which of its immediate children to instantiate next.

**Table 2**  
Custom elements for AA materials.

Element name	Description
aa-text-answer	A textfield for free text input.
aa-multiple-choice	A group of choices presented as radio buttons, to choose a single one from.
aa-checkboxes	A group of choices presented as checkboxes, to choose many from.
aa-choice-item	A choice within either aa-multiple-choice or aa-checkboxes.
aa-likert-scale	Likert scale container, essentially a specific configuration of aa-multiple-choice.
aa-geolocation	Retrieves the user's GPS location.
aa-screen	Groups contents together and provides a single "submit" button for the user to signal that this particular request for data was completed.

applications or hybrid mobile apps (i.e., Web-apps that also invoke custom-made functionality in the native host platform) for researchers to conduct AA studies with. In previous work [6] we sought to take advantage of the potential to run AA studies either exclusively, or mainly in the browser, while allowing the composition of the protocol as a program. Results showed that the approach could suitably serve actual use cases for research [7], and gave rise to the question of whether the whole protocol could be represented as HTML. While it is generally straightforward to write the materials of an AA study in HTML, there is no support for procedural logic in HTML scripts.

We contribute therefore a set of custom HTML elements that help treat parts of an HTML document as an executable program. Also, the provision of additional custom elements, for easier declaration of commonly used materials, helps raise the level of abstraction from the syntactic elements of the web-page, to the semantics of the AA study. The result is a formal, unambiguous representation of an AA protocol, in both its materials and procedures, that is human-readable and computer-executable, and serves as the single source for both the description and the implementation of an AA study, and which can be of use in the exact dissemination and replication of AA research protocols [8].

## 2. Software description

This software is a collection of custom HTML elements, which are made possible by Web Components [9], a set of technologies in the HTML5 specification. Web Components allow users to define their own HTML tags, which when parsed by the browser will be instantiated into the corresponding custom elements. A custom element's definition consists of HTML and CSS that the custom element encapsulates, and a set of callback functions in

JavaScript for the browser to execute at different stages of the element's life-cycle, such as when it is appended to the document, or when one of its attributes is changed.

The present set of custom elements is written to achieve 3 objectives:

- Ability to inject sub-trees of HTML nodes into the document after the document has been initially parsed. A typical approach to presenting information partially or progressively is, after allowing the browser to instantiate all possible nodes, hide some in order to make them visible later in response to user interactions. In our case, content instantiation and attachment to the document must take place only at opportune times, as custom elements can be proxies for various tasks (e.g., sending an HTTP request to a server, or initiating a function of the hardware such as GPS location retrieval).
- Prescribe how injection happens, e.g. in a synchronous sequence, conditionally, etc. The author should be able to define the flow of execution. To allow for the general case, the software needs to provide sequential execution, conditional branching, read/write memory operations, and the ability to jump to different points in the program sequence. These purposes are served by the custom elements outlined in Table 1.
- Provide HTML tags that use semantics of the domain activity, and reduce the use of syntactic elements of layout. In order to make an HTML document easier to understand for domain experts, it serves to use terms that match the language of the application domain. Table 2 lists an indicative subset of content and User Interface oriented elements that encapsulate HTML syntax and use tag names descriptive of their utility in a AA protocol.

### 2.1. Software architecture

Fig. 1 presents an overview of the custom elements that make up the software release and how they relate to each other. All elements inherit from aa-base-element, which provides them with a common set of functions. Generally, execution happens as follows:

1. When attached, aa-session will look at the tag names of its immediate children. If any of them belong to the element set, it will remove their inner HTML and place it in a document fragment object that is a member of that element. This prevents their content from being instantiated, so each can decide how to process it individually.
2. Most other elements, when attached to the DOM, will scan their given document fragments and perform the same replacement with their own immediate children. Each will then take appropriate action selecting and attaching content, as per their functionality. For example, aa-session will attach a copy of one child per its execution cycle (Fig. 2a), while aa-choose will only attach copies of its aa-when children whose test condition evaluates to true (Fig. 2b). Making copies into the DOM instead of moving nodes out of the document fragment and into the DOM, allows these elements to execute multiple times and keep produce new content into the document if necessary.
3. Elements such as aa-choose, aa-jump or aa-variable will also dispatch an "end" event, which aa-sequence listens for. Upon reception, it will attach a copy of its next child, and sequential execution will ensue. Elements such as aa-sequence do not dispatch such an event, but allow the User Interface to decide when that should be done, e.g. once the user has pressed a "submit" button for having answered all questions of a questionnaire.

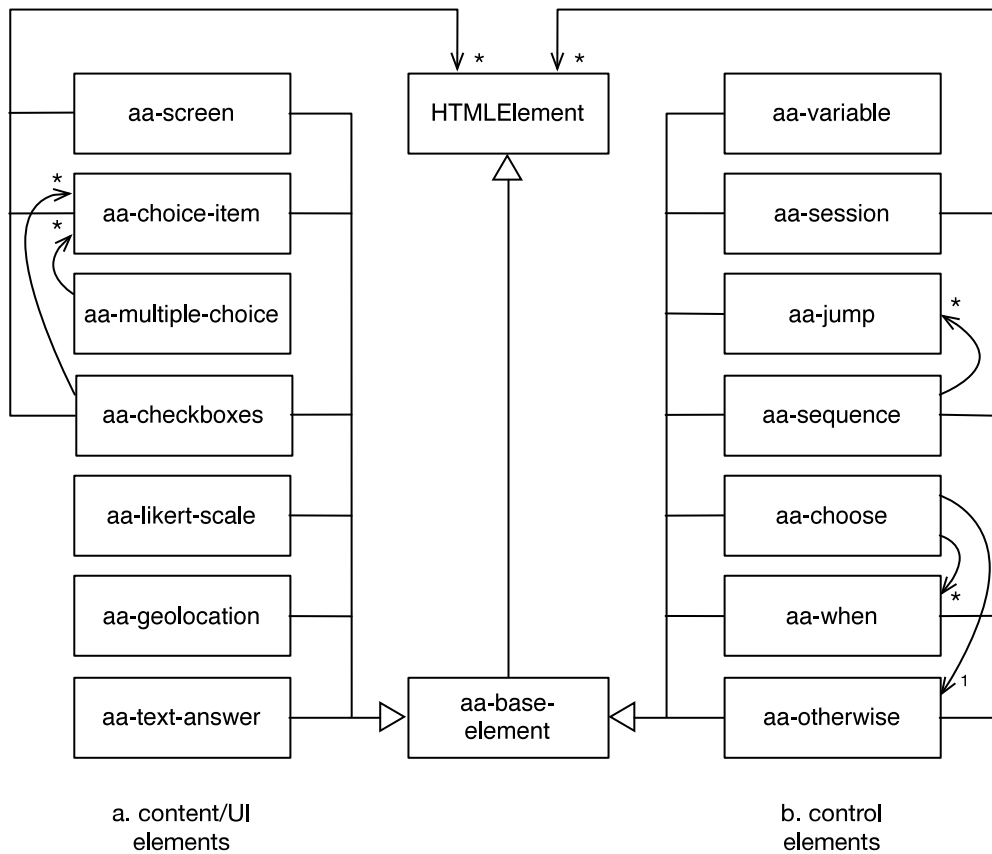


Fig. 1. The custom elements are derived from a subclass of HTMLElement, aa-base-element, which provides them with a common set of methods for treating their contents.

4. Elements that generate data (either captured from user input or other sources), will include it into the dispatched “end” event which bubbles up to the aa-session element. In its turn, it can send a HTTP POST request to a URI that is assigned to it via a “post-to” attribute.

### 3. Illustrative examples

#### 3.1. 10 PRINT CHR\$(205.5+RND(1)); : GOTO 10

This famous Commodore64 BASIC oneliner [10] randomly generates either \ or / and prints the character in an endless loop, producing a maze-like image. Reproducing it let us demonstrate variable writing and reading, sequential and conditional execution, and jumping in executable HTML. An equivalent program is shown in Listing 1, and Fig. 3 displays the resulting output.

#### 3.2. Randomized part of a questionnaire

There are cases where varying items on a questionnaire encourages a participant’s engagement with the material, while evaluating the same underlying constructs. In a previous study [11] participants had been asked to report on different aspects of their social interactions, with such a varying component in place. Listing 2 shows the construction of such an example for the purposes of a questionnaire, where one out of three possible aa-screen elements is selected, and waits for user input to proceed.

### 4. Impact

HTML is typically used as a language for structuring information, in a way that Web-browsers can interpret and construct a visual presentation of it for the user to peruse. Designing it, Tim Berners-Lee advocated for the rule of least power [12], according to which, the less powerful the language that is chosen for a solution is, the more easy it is to analyze and reuse the data represented in that language. Therefore HTML was intentionally designed to not be a full programming language, but meant instead for the declaration of informational materials and their syntactic features, so that software that parses HTML can perform different operations with it, such as render it in different ways, or extract specific types of information from it.

However, it is also important to facilitate users’ interactions with the information in Web documents. In modern Web-browsers, while document content is expressed in HTML, interactions and procedural tasks are specified in the JavaScript programming language. Ultimately, web-document authors need to operate in the modalities of both HTML and JavaScript, which can be difficult for those who do not operate with a JavaScript programmer’s mindset, and would find it easier to understand and manipulate HTML.

In the field of clinical psychology, use-cases in which this software is addressing so far, end-user-friendly ways for constructing computerized protocol materials are always desirable [13]. Additionally, efforts to catalog and formalize protocol components are underway [14].

HTML is a way to represent this type of information, that is both human-readable and formally tractable [15]. The custom tag names offered here can help a document’s composition to that of Executable HTML contributes a procedural component, which

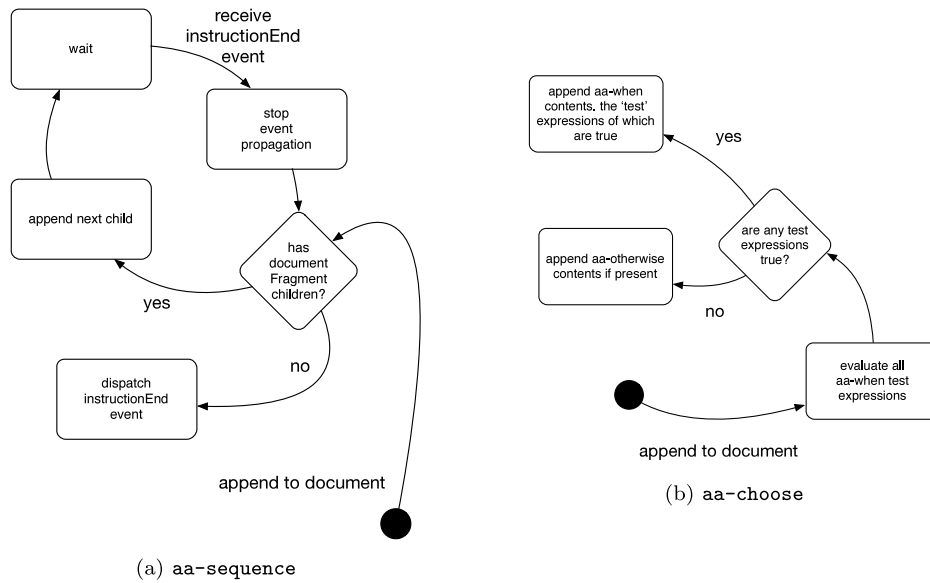
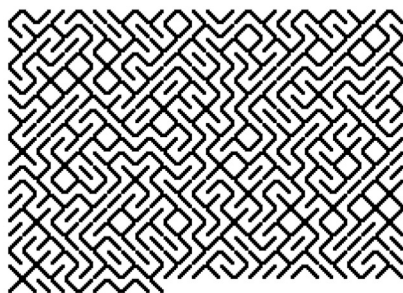


Fig. 2. Execution flows for aa-sequence and aa-choose.

```
<aa-session><template>
  <aa-sequence>
    <aa-function-random min="1" max="2" name="rnd">
    </aa-function-random>
    <aa-choose>
      <aa-when test="rnd==1">m</aa-when>
      <aa-otherwise>n</aa-otherwise>
    </aa-choose>
    <aa-jump goto="rnd"></aa-jump>
  </aa-sequence>
</template></aa-session>
```

Listing 1: 10 PRINT CHR\$(205.5+RND(1)); : GOTO 10. A `template` element as child of `aa-session` prevents immediate execution of both constructors and callback functions of nested elements. This ensures that none of our custom-element code is executed upon document parsing by the browser.



(a) rendering characters 'm' and 'n' in the C64 typeface produces a maze-like visual output

```
<div class="c64char">
  <aa-session>
    <aa-sequence class="democontainer" type="elements">
      <stopped="true">
        <aa-function-random min="1" max="2" name="rnd">
        </aa-function-random>
        <aa-choose>
          <aa-when test="rnd==1">m</aa-when>
          <aa-otherwise>n</aa-otherwise>
        </aa-choose>
        <aa-jump goto="rnd"></aa-jump>
      </aa-sequence>
    </div>
```

(b) HTML produced by the executable custom elements. Control elements `aa-function-random`, `aa-choose`, `aa-jump` remove themselves from the final result and only inject copies of the text-nodes.

Fig. 3. Output of Listing 1.

allows a protocol author to describe, alongside materials, also the logic of their presentation to a study participant. In the AA field, the proposed method can be impactful in the following ways:

- It can reduce, or for common scenarios eliminate the need to write in a programming language, and make the distribution and reproduction of data collection protocols more accessible to researchers. Furthermore, there is abundance of tools that support the composition and publication of HTML for a

very wide range of user-expertise, and can be leveraged to also support the composition of custom HTML elements.

- The same document can be just as easily treated as data instead of input to the browser. Comparisons between protocols can be made in automated ways, and matched against the data they help collect. For example, different versions of a protocol document can be contrasted against the compliance of participants each one encourages.



```

<aa-sequence id="s1">
  <aa-function-random min="1" max="3" name="random_choice"></function-random>
  <aa-choose>
    <aa-when test="random_choice==1">
      <aa-screen>
        Did you do any of the following?
        <aa-multiple-choice name="behaviours1">
          <aa-choice-item value="1"> I listened attentively to the other(s)</aa-choice-item>
          <aa-choice-item value="2">I tried to get the other(s) to do something else</aa-choice-item>
          <aa-choice-item value="3">I let other(s) make plans or decisions</aa-choice-item>
        </aa-multiple-choice>
      </aa-screen>
    </aa-when>
    <aa-when test="random_choice==2">
      <aa-screen>
        Did you do any of the following?
        <aa-multiple-choice name="behaviours2">
          <aa-choice-item value="1">I made a concession to avoid unpleasantness</aa-choice-item>
          <aa-choice-item value="2">I did not state my own views</aa-choice-item>
          <aa-choice-item value="3">I expressed the 'real me' instead of my public persona</aa-choice-item>
        </aa-multiple-choice>
      </aa-screen>
    </aa-when>
    <aa-otherwise>
      <aa-screen>
        Did you do any of the following?
        <aa-multiple-choice name="behaviours3">
          <aa-choice-item value="1">I took the lead in planning an activity</aa-choice-item>
          <aa-choice-item value="2">I avoided taking the lead</aa-choice-item>
          <aa-choice-item value="3">I ignored the other(s) comments</aa-choice-item>
        </aa-multiple-choice>
      </aa-screen>
    </aa-otherwise>
  </aa-choose>
</aa-sequence>

```

Listing 2: A parent sequence with two nested children. The first child is a random number generator storing values into the variable `random_choice`. After the first child finishes execution (i.e. generates and saves the value), the second child is instantiated. It is a `choose-when-otherwise` block that tests the variable `random_choice` for its value and conditionally instantiates one of three multiple choice groups, where the user is asked to indicate a behaviour they have exhibited.

- It is platform-independent and not specific to any particular application framework or software product, proprietary or otherwise. Its adoption can reduce the burden of producing AA tools by being a software component, and encourage interoperability amongst software from different vendors as a protocol import/export format.

## 5. Conclusions

This paper contributes a way to treat HTML content as statements in an imperative program, by way of custom HTML elements that instantiate and attach their child-nodes in the HTML document. Motivation for this work stems from the development of a software system for the definition of ambulatory assessment protocols in psychology (e.g., to repeatedly survey participants in situ, over a sustained period). In this field, there is opportunity to serve the reproducibility of research studies by detailing procedures and materials in formal, unambiguous representations, rather than the natural language that is predominant in current literature.

While the work originates in the context of AA studies, it can potentially be of use in other applications where end-users can make use of HTML to specify programmed behavior without having to write code in JavaScript, or reduce the extent to which they have to do so.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] Conner TS, Mehl MR. Ambulatory assessment: Methods for studying everyday life. In: *Emerging Trends in the Social and Behavioral Sciences: An Interdisciplinary, Searchable, and Linkable Resource*. Wiley Online Library; 2015.
- [2] Vogt WP, Johnson RB. *Dictionary of statistics & methodology: a nontechnical guide for the social sciences: a nontechnical guide for the social sciences*. Sage; 2011.
- [3] Stone AA, Shiffman S. Capturing momentary, self-report data: A proposal for reporting guidelines. *Annals Behav Med* 2002;24(3):236–43. [http://dx.doi.org/10.1207/S15324796ABM2403\\_09](http://dx.doi.org/10.1207/S15324796ABM2403_09).
- [4] Stone A. Challenges remaining for the field of real-time data capture. 2017, URL <https://cesr.usc.edu/documents/SAA>.
- [5] Ince DC, Hatton L, Graham-Cumming J. The case for open computer programs. *Nature* 2012;482(7386):485.
- [6] Batalas N, Markopoulos P. Considerations for computerized in situ data collection platforms. In: *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. EICS '12, New York, NY, USA: ACM; 2012, p. 231–6. <http://dx.doi.org/10.1145/2305484.2305522>, URL <http://doi.acm.org/10.1145/2305484.2305522>.
- [7] Batalas N, aan het Rot M, Khan VJ, Markopoulos P. Using TEMPEST: End-user programming of web-based ecological momentary assessment protocols. *Proc ACM Hum Comput Interact* 2018;2(EICS). <http://dx.doi.org/10.1145/3179428>.

- [8] Batalas N, Khan V-J, Franzen M, Markopoulos P, aan het Rot M. Formal representation of ambulatory assessment protocols in HTML5 for human readability and computer execution. *Behav Res Methods* 2019;51(6):2761–76. <http://dx.doi.org/10.3758/s13428-018-1148-y>.
- [9] W3C. Web components. 2016, Accessed: 2020-01-28, <https://github.com/w3c/webcomponents/>.
- [10] Montfort N, Baudoin P, Bell J, Bogost I, Douglass J, Marino MC, Mateas M, Reas C, Sample M, Vawter N. 10 PRINT CHR \$(205.5+ RND (1));: GOTO 10. MIT Press; 2012.
- [11] Bosman RC, Albers CJ, de Jong J, Batalas N, aan het Rot M. No menstrual cyclicity in mood and interpersonal behaviour in nine women with self-reported premenstrual syndrome. *Psychopathology* 2018;51(4):290–4.
- [12] Berners-Lee T, Mendelsohn N. The rule of least power. 2006, URL <https://www.w3.org/2001/tag/doc/leastPower.html>.
- [13] Geisler-Brenstein E, Brenstein RJ. The potential of hypercard for psychological research and instruction: A general discussion and description of two research applications. *Behav Res Methods Instrum Comput* 1989;21(2):307–11.
- [14] Kirtley O, Hiekkaranta A, Kunkels Y, Verhoeven D, Van Nierop M, Myin-Germeys I. The experience sampling method (ESM) item repository. *Open Sci Framework* 2019. <http://dx.doi.org/10.17605/OSF.IO/KG376>, URL <https://osf.io/kg376/>.
- [15] Khare R, Rifkin A. The origin of (document) species. *Comput Netw ISDN Syst* 1998;30(1):389–97.