An Integrated Model for Content Management, Topic-oriented User Segmentation, and Behavioral Targeting

Hans-Werner Sehring
Namics – A Merkle Company
Hamburg, Germany
e-mail: hans-werner.sehring@namics.com

Abstract—**The World Wide Web is the basis for increasingly many information and interaction services. Personalization provides users with information and services that are adequately tailored to their current needs. Targeting, a form of implicit personalization of content and content presentations for groups of users, comes to broader practical use for a growing number of commercial websites. The wider adoption results from the availability of platforms that incorporate targeting. Solutions are usually built on top of content management systems or eCommerce solutions that are used for the production of websites. The rule sets that typically are required for targeting are related to content, but they are superimposed in the sense that they are not an integral part of the content model or the content itself. This paper presents an initial model that is used to study the integration of content, content visualizations, user classification, and content targeting. Potential benefits from an integrated model are manifold. In the presented approach, personalization is applied by putting content in context rather than through superimposed targeting rules. By expressing personalization rules in the same context-dependent and evolvable way as content, they can also evolve over time and can be adapted to different user contexts. On top of that, they can be defined and maintained by content editors and other users of a content management system. The models used to study this form of integrated targeting do not rely on a certain technology or implementation. The features of the Minimalistic Meta Modeling Language (M3L) are employed as a meta model testbed, though. It allows expressing personalization rules along with both the content they refer to and the concepts they are based on, as well as users' interest, in the required consistent way.**

Keywords-personalization; targeting; segmentation; web tracking; profiling; content management.

I. INTRODUCTION

The World Wide Web has undergone a tremendous development. For over two decades now, there is research on *personalization* of contents that are published on the web and of the presentations used for its publication.

For quite some time now, personalized content publication has arrived in the practical operation of websites. *Targeting*, the form of personalization that is found in practice most often, has been studied in [1]. In this article, we extend this work by modeling and analyzing the various deduction steps that are involved in targeting in more depth.

There is a wide range of personalization approaches for different purposes and goals [2]. These approaches differ in several aspects [3], e.g., in the way personalizations are derived: explicitly by users stating their preferences or implicitly by deriving them from users' behavior and habit. An example for explicit personalization are websites that allow users to name their interests or that allow to individually rearrange parts of the web site. Implicit personalization is achieved, e.g., by observing interactions of a user with a website [4] or by taking previously visited websites into consideration (customer journeys, at best).

Targeting that is based on user behavior is centered around the assumption that users' interest can be derived from their search or browsing behavior [5].

Personalization approaches also differ in the subject of the individual adaptations, e.g., content or content representations (visualizations of content created for publication). Content personalization can be found, e.g., in online shops where users receive individual pricing. Content visualizations are personalized by, e.g., ordering lists of content entries in a user-specific way.

Personalization has already been adopted to a range of specific, innovative websites, in particular those that confront the user with large amounts of content [6][7]. Such websites use personalization to filter and prioritize content based on assumed user preferences.

Currently, *targeting* is applied by an increasing number of commercial websites. We consider targeting as implicit personalization of content for user groups. The adaptation of content is limited to predefined points, though. Typically, part of the content is selected from building blocks that are prepared for the different user groups.

A set of tools that has emerged during the past years constitutes the basis that allows configuring websites for personalization. Examples are personalization engines built into content management systems and commerce platforms, as well as external personalization services that allow adjusting websites to specific user groups.

There is a lack of models that would cover multiple kinds of personalization approaches [8] and, therefore, allow different usage scenarios to be integrated in one solution.

Typically, commercial products use means of personalization that are superimposed to a (non-individualized) base system. A content management system, e.g., allows defining a content model according to which content will be edited, managed, and published. This content model is defined in a uniform way for all users and application scenarios. On a different layer, personalization is added by other means, typically rules that define how to adjust content representations of specific user groups.

Therefore, there is no coherence between content models, content visualization layouts, and personalization rules in such systems. Instead, content has to be defined with all possible audiences and usage scenarios in mind, visualizations have to provide the variations to be offered as personalizations, and personalization rules may only be defined within the limits set by these definitions.

Contemporary products typically require fixed content models and visualizations (or at least ones that cannot be changed by content editors). This only leaves such personalization rules at the content editors' disposal that can be defined with respect to the possibilities and constraints raised by content models and content visualizations.

The aim of this paper is providing first studies towards a fully integrated model that combines many aspects of content and its personalized utilization. For this study, we use the Minimalistic Meta Modeling Language (M3L) [9] as a testbed. This language is well-suited for content models since it covers variations and contexts of content in a direct way. Insights into a variety of personalization options originate from previous research on Concept-oriented Content Management [10]. These insights are transferred to M3L models.

The rest of this paper is organized as follows. Section II describes targeting approaches typically found in commercial software products. Section III provides a short introduction into the M3L. Section IV gives a brief outlook on an architecture of M3L implementations. Sections V to XI present the modeling experiment of utilizing the M3L for expressing and integrating the common targeting approach into content models for websites. Conclusions and acknowledgement close the paper.

II. Targeting in Commercial Products

There is a wide range of approaches to personalization that can be found in the literature and in prototype implementations. In this paper, we constrain ourselves to targeting, which is of particular importance for commercial websites. Targeting is a form of implicit personalization of content assembled for presentation with respect to a customer group. The personalization itself is directed by rules set up by content editors.

Another application of targeting is advertising, also used for so-called retargeting. In contrast to website personalization, targeted advertising allows to quantitatively measure success. In many cases, targeting turned out to significantly increase advertising success [11].

A. Segment-based Targeting Rules

For targeting, as it is found in many commercial products, users of a web site are assigned *segments*. Segments are categories describing a user's interest or preferences. These segments are predefined for a particular website [12] (though there are scientific approaches that include deriving segments by, e.g., means of clustering [13]). Segments connect website topics with user preferences.

The assignment of segments to users is based on *tracking* (or *analytics*) used during web page delivery. By tracking, accesses to web pages are recorded. Tracking can be integrated into the system that delivers the web pages, or it can be performed by an external service [14]. Depending on the granularity required, interactions on smaller parts than whole pages may be counted [15].

From the web pages visited by a user, her or his interests are derived by collecting the topics covered by those web pages. The web pages considered in this collection could be, e.g., those web pages that have been visited most often, or the web pages for which the visits exceed a given threshold.

The segments assigned to a user (by that time) are used as a parameter to content selection and to the production of documents from content. This way, content and its representations are personalized for user groups, namely groups consisting of users with the same segments assigned.

B. Related Work

Targeting is found in diverse systems and services, e.g., in Content Management Systems (CMSs), commerce systems, and marketing suites.

All commercial approaches are limited by the fact that certain parts of the targeting process are built into the solution. The means for producing targeted content are given by the software and typically consist of rules that select content based on certain parameters. Segments often cannot be changed dynamically, but are predefined during system configuration. The approach of this article aims to overcome these restrictions and to make all parts of the targeting process accessible to editors.

1) Personalization Engines in Content Management Systems. Some CMSs have means of segmentation built in. These systems allow equipping content with rules for the selection of content to be included in published web pages based on user segmentation. These rules are applied during document production. Like many others, the CMS products of CoreMedia [16] and Sitecore [17] work this way.

2) Superimposed Personalization. Instead of an integrated personalization engine inside a CMS, an external service can alternatively perform personalization. Adobe Target [18] is a prominent representative of this personalization approach. External tracking is based on published documents, not the content itself. Likewise, targeting is performed by modifying documents.

3) Consideration of Additional Information on Users. Instead of just considering user behavior in the form of web page access profiles, increasingly many applications are also based on explicit customer data. Such data come from, e.g., a Customer Relationship Management (CRM) system, from the history of transactions in a commerce system, from the history of cases in a support system, or from feedback given by means of ratings. Personalization may additionally be based on context information, e.g., the time of day, the device the visitor uses, or some kind of work mode she or he is in [19]. Such context information is partially considered in commercial personalization engines.

III. The Minimalistic Meta Modeling Language

The *Minimalistic Meta Modeling Language* (*M3L*, pronounced "mel") is a modeling language that is applicable to a range of modeling tasks. It proved particularly useful for context-aware content modeling [20].

For the purpose of this paper, we only introduce the static aspects of the M3L in this section. Dynamic evaluations that are defined by means of different rules are presented in the subsequent section.

The descriptive power of M3L lies in the fact that the formal semantics is rather abstract. There is no particular (domain) semantics of M3L concept definitions. There is also no formal distinction between typical conceptual relationships (specialization, instantiation, entity-attribute, aggregation, materialization, contextualization, etc.).

A. Concept Definitions and References

A M3L definition consists of a series of definitions or references. Each definition starts with a previously unused identifier that is introduced by the definition and may end with a period, e.g.:

```
Person.
```

A reference has the same syntax, but it names an identifier that has already been introduced.

We call the entity named by such an identifier a *concept*.

The keyword `is` introduces an optional reference to a *base concept*, making the newly defined concept a *refinement* of it.

A specialization relationship as known from object-oriented modeling is established between the base concept and the newly defined derived concept. This relationship leads to the concepts defined in the context (see below) of the base concept to be visible in the derived concept.

The keyword `is` always has to be followed by either `a`, `an`, or `the`. The keywords `a` and `an` are synonyms for indicating that a classification allows multiple sub-concepts of the base concept:

```
Peter is a Person. John is a Person.
```

There may be more than one base concept. Base concepts can be enumerated in a comma-separated list:

```
PeterTheEmployee is a Person, an Employee.
```

The keyword `the` indicates a closed refinement: there may be only one refinement of the base concept (the currently defined one), e.g.:

```
Peter is the FatherOfJohn.
```

Any further refinement of the base concept(s) leads to the redefinition ("unbinding") of the existing refinements.

Statements about already existing concepts lead to their redefinition. For example, the following expressions define the concept Peter in a way equivalent to the above variant:

```
Peter is a Person.
Peter is an Employee.
```

```
Person {
  Name is a String.
}
Peter is a Person{
  "Peter Smith" is the Name.
}
Employee {
  Salary is a Number.
}
Programmer is an Employee.
PeterTheEmployee is a Peter, a Programmer{
  30000 is the Salary.
}
```

Figure 1. Simple M3L definitions.

B. Content and Context Definitions

Concept definitions as introduced in the preceding section are valid in a context. Definitions like the ones seen so far add concepts to the top of a tree of contexts. Curly brackets open a new context. Figure 1 shows a sample M3L model.

We call the outer concepts the *context* of the inner, and we call the set of inner concepts the *content* of the outer.

In this example, we assume that concepts **`String`** and **`Number`** are already defined. The sub-concepts created in context are unique specializations in that context only.

As indicated above, concepts from the context of a concept are inherited by refinements. For example, Peter inherits the concept **`Name`** from **`Person`**.

M3L has visibility rules that correlate to both contexts and refinements. Each context defines a scope in which defined identifiers are valid. Concepts from outer contexts are visible in inner scopes. For example, in the above example the concept **`String`** is visible in **`Person`** because it is defined in the topmost scope. **`Salary`** is visible in **`PeterTheEmployee`** because it is defined in **`Employee`** and the context is inherited. **`Salary`** is not valid in the topmost context and in **`Peter`**.

Concepts that are not visible in one scope because they are only defined in a neighboring scope can be accessed with the `from` clause. For example, the **`Name`** of **`Peter`** is accessed by

```
Name from Peter.
```

C. Contextual Amendments

Concepts can be redefined in contexts. This happens by definitions as those shown above. For example, in the context of **`Peter`**, the concept **`Name`** receives a new refinement.

Different aspects of concepts can explicitly be redefined in a context, e.g.:

```
AlternateWorld {
  Peter is a Musician {
    "Peter Miller" is the Name.
  }
}
```

We call a redefinition performed in a context different from that of the original definition a *conceptual amendment*.

```
Person {
  Sex.
  Status.
}
MarriedFemalePerson is a Person {
  Female is the Sex.
  Married is the Status.
}
MarriedMalePerson is a Person {
  Male is the Sex.
  Married is the Status.
}
Mary is a Person {
  Female is the Sex.
  Married is the Status.
}
```

Figure 2. Concept refinements and a potential narrowing.

In the above example, the contextual variant of Peter in the context of **AlternateWorld** is both a **Person** (initial definition) and a **Musician** (additionally defined). The **Name** of the contextual **Peter** has a different refinement that replaces the original one.

A redefinition is valid in the context it is defined in, in sub-contexts, and in the context of refinements of the context (since the redefinition is inherited as part of the content).

D. *Concept Narrowing*

There are three important relationships between concepts in M3L.

M3L concept definitions are passed along two axes: through visibility along the nested contexts, and through inheritance along the refinement relationships.

A third form of concept relationship, called *narrowing*, is established by dynamic analysis rather than by static definitions like content and refinement.

For a concept c_1 to be a narrowing of a concept c_2, c_1 and c_2 need to have a common ancestor, and they have to have equal content. Equality in this case means that for each content concept of c_2 there needs to be a concept in c_1's content that has an equal name and the same base classes.

For an example, consider the definitions **MarriedFemalePerson** and **MarriedMalePerson** shown in Figure 2.

With these definitions, the concept **Mary** is a narrowing of **MarriedFemalePerson**, even though it is not a refinement of that concept, and though it introduces separate nested concepts **Female** and **Married**:

- They have the common ancestor **Person**.
- For each concept in the content of **MarriedFemalePerson** (**Female**, **Married**) there is a concept in **Mary** with the same name and common base classes

Mary is not a narrowing of **MarriedMalePerson** since it lacks a concept named **Male** in its content.

E. *Semantic Rule Definitions*

For each concept, one *semantic rule* may be defined.

```
MarriedFemalePerson is a Person {
  Female is the Sex.
  Married is the Status.
} |= Wife.
MarriedMalePerson is a Person {
  Male is the Sex.
  Married is the Status.
} |= Husband.
```

Figure 3. Concept redefinitions.

The syntax for semantic rule definitions is a double turnstile ("|=") followed by a concept definition. A semantic rule follows the content part of a concept definition, if such exists.

A concept definition within a semantic rule is not made effective directly, but is used as a prototype for a concept to be created later.

The concepts shown in Figure 3 redefine the concepts **MarriedFemalePerson** and **MarriedMalePerson**.

The concepts **Wife** and **Husband** are not added to the model directly, but at the time when the parent concept is evaluated. Evaluation is covered by the subsequent section.

Concepts from semantic rules are created and evaluated in different contexts. The concept is instantiated in the same context in which the concept carrying the rule is defined. The context for the evaluation of a rule (evaluation of the newly instantiated concept, that is) is that of the concept for which the rule was defined.

In the example above, the concept **Wife** is created in the root context and is then further evaluated in the context of **MarriedFemalePerson**.

Rules are passed from one concept to another by means of inheritance. They are passed to a concept from (1) concepts the concept is a narrowing of, and (2) from base classes. Inheritance happens in this order: Only if the concept is not a narrowing of a concept with a semantic rule then rules are passed from base concepts.

For example, **Mary** as defined above evaluates to **Wife**.

F. *Syntactic Rule Definitions*

Additionally, for each concept one *syntactic rule* may be defined.

```
" ". ".".
WordList {
  Word.
  Remainder is a WordList.
} |- Word " " Remainder.
OneWordWordList is a WordList |- Word.
Sentence { WordList. } |- WordList ".".
HelloWorld is a Sentence {
  Words is the WordList {
    Hello is the Word.
    OneWordWordList is the Remainder {
      World is the Word.
    }
  }
}
```

Figure 4. Syntactic rules of M3L concepts.

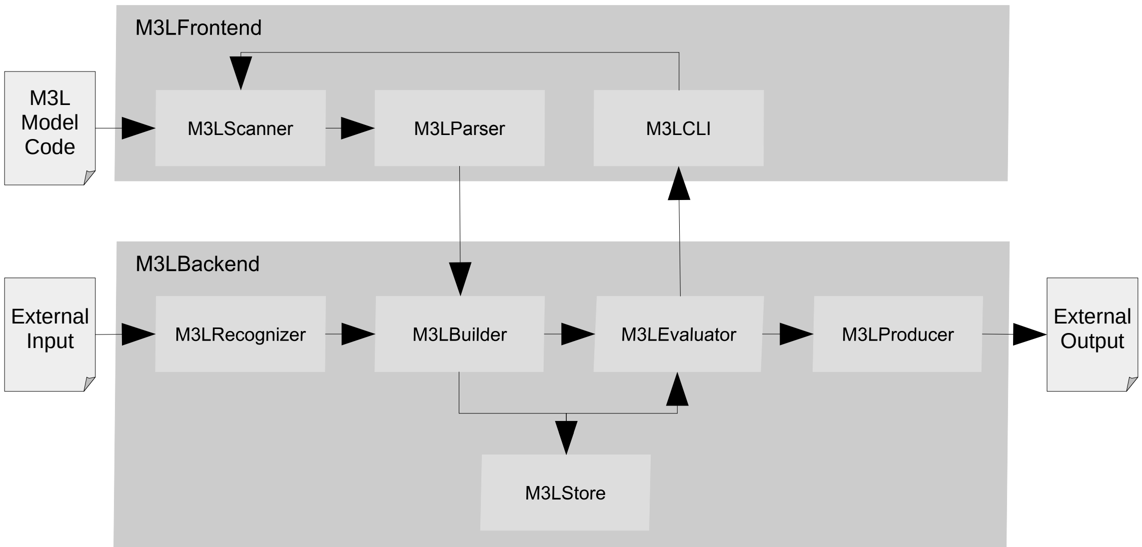

Figure 5. High-level architecture of the M3L system framework.

Such a rule, like a grammar definition, can be used in two ways: to produce a textual representation from a concept, or to recognize a concept from a textual representation.

A semantic rule consists of a sequence of concept references or **`the name`** expressions that evaluate to the current concept's name.

During evaluation of a syntactic rule, rules of referenced concepts are applied recursively. Concepts without a defined syntactic rule are evaluated to/recognized from their name.

As an example, from the definition of the concept **`HelloWorld`** in Figure 4, the textual representation `Hello World.` is produced. In this example, first the two concepts named blank (" ") and dot (".") are defined. These are simply used as string literals.

The concept **`WordList`** represents a simple linked list with entries containing one word and a reference to the next entry. The concept is defined with a syntactical rule that consists of a sequence of the concepts **`Word`** (the word in sequence), " ", and **`Remainder`** (the next entry).

The concept **`OneWordWordList`** represents lists with just one entry, and it overrides the syntactical rule with the respective definition. A sentence is a **`WordList`**, syntactically followed by a full stop (".").

`HelloWorld` is a concrete **`Sentence`**. When its syntactic form is produced, the rule inherited from **`Sentence`** is used. This one delegates text production to the rule of **`WordList`** and adds a full stop. **`Words`** in the content of **`HelloWorld`** inherits this rule and prints out the words (**`Hello`** and **`World`**) in sequence, separated by the blank. The concept for the blank has no syntactic rule, so it prints out its own name, thus producing the whitespace.

IV. AN ARCHITECTURE FOR M3L IMPLEMENTATIONS

A system for the management and evaluation of M3L models can be implemented in a straightforward way. Current work towards a M3L system implementation aims at a framework within which different system configurations can be set up.

The high-level architecture of that framework is sketched in Figure 5. The rectangle boxes represent classes of components that are filled with matching implementations in an actual system configuration. The arrows denote the flow of model representations.

The basic architecture follows that of a classical compiler. It can be separated into a frontend and a backend part. The frontend part is mainly comprised of a scanner and a parser, which read in M3L code and produce an intermediate representation of the M3L models from that code.

The frontend can optionally host a command line interface (CLI). Concrete interfaces allow evaluating lists of concept references (thus triggering their semantic rules), or they provide interactive operation by directly applying M3L statements, which a user typed in.

A *M3LBuilder* component constitutes the interface to the backend. A builder receives the intermediate representation of M3L statements and builds up the model structures in the *M3LStore*, a component that abstracts from model storage.

Another central component of the backend is a *M3LEvaluator*. This component resolves concept references in contexts and it evaluates semantic rules of the concepts. It operates on the *M3LStore*.

A *M3LStore* manages concepts that have explicitly been defined (created by a *M3LBuilder*) and ones that have been produced from rules (by a *M3LEvaluator*). A simple implementation of a store for the management of models operates in main memory. Others persist concepts in an external database system [20].

The *M3LBuilder* and in particular the *M3LEvaluator* query the store a lot when resolving concepts in their contexts. Because of the different query capabilities of different storage technologies, the framework allows optimized implementations of the builder and the evaluator that utilize the underlying store best. For example, there are different strategies to resolve concepts in the two tree structures of concept refinement and context containment.

```
WebPage.
String. Integer. FormattedText.
ContentReference.
Teaser.
OverviewPage is a WebPage {
  Title is a String.
  MainContent is a String.
  NewsTeaser is a Teaser.
}
```

Figure 6. Base model for targetable websites.

There are two components dedicated to syntactical rules: recognizers and producers. These interface with the environment.

Producers are typically generated from syntactic rules by transforming them into a grammar definition. A grammar definition is then processed by a parser generator. The generation of grammars from syntactic rules became feasible with the advent of powerful tools such as AntLR [21].

For recognizers, templates for the production of text are generated. Template generation is feasible since concept evaluation is done by the evaluator. The templates only contain code to substitute concept names with the output of the evaluation of the referenced concepts (compare [22]). That output is in turn produced by templates.

There are multiple possible generators for recognizers and producers that operate on different input and output channels. This allows specific implementations for applications.

For example, for web-based systems, the recognizer may interface with a web server to analyze HTTP requests that are directed at concepts. A request to *http://server/A/B/C* resolves `C from B from A` in the M3L. The producer can be a templating engine that creates external output to be transmitted as the response to the HTTP request.

V. A Model of Content Personalization

The M3L allows defining structured content models, as well as content itself. Furthermore, it can be used to define layouts, for example, for web pages that define how content is rendered into documents.

In this article, we demonstrate how to add targeting to such models. The aim is to allow integrated models for content, layout, segmentation, and personalization.

Such an integrated model allows content editors to define segmentation as well as targeting rules together with content. This way, they are not restricted to a set of predefined rules for content selection for rendering, but have full modeling power. All models are put under control of the editors. They can, for example, synthesize content, change the layout, even add new renderings, etc.

With segmentation and targeting integrated in content models, all definitions are made in a coherent and consistent way. Since the difference between a content model, content, and content personalization in the M3L is merely a matter of contextual interpretation, there is much modeling liberty in setting up targeted content (presentations).

The subsequent sections incrementally introduce one way of setting up such structures. provide a first simple M3L model of content, its visualization on web pages, website users, web page accesses, and the targeting of the web pages to the users based on past web page accesses. The sections follow the principles of targeting as follows:

- Section VI covers basic website content management and web page rendering.
- A classification of users through segments is introduced in Section VII.
- Section VIII extends the basic content management by introducing targetable websites.
- Classical web tracking is modeled in M3L in Section IX.
- Means to turn track records into a segmentation of users, here based on users' interest, are presented in Section X.
- Finally, Section XI discusses the utilization of segments to apply targeting rules on targetable websites.

Every section gives base definitions required to add a modeling feature to content management, as well as sample applications by which we can discuss these features.

VI. A Web Page Model

This section demonstrates simple content, content presentation, and content delivery models for a basic website. For the integrated models developed in the course of this article, we formulate these models in the M3L. On that basis, the subsequent sections add targetable website definitions, tracking, segmentation, and finally targeting.

A. Base Definitions for Website Structure and Content

The M3L allows modeling content in itself. Therefore, no specific base model is required for content alone. For the models to be integrated, some structures are defined, though.

The sample model in Figure 6 starts with the concept `WebPage` that constitutes the root of the content model. The concept itself defines no further structure. Is serves as a base concept for refinements that contain content and that syntactically evaluate to HTML code.

From a pure content management perspective, a web page is not a proper piece of content. Content should be independent of a publication channel, layout, etc. For the sake of simplicity, though, we take this simple model as a shortcut to our examples.

In many contemporary content management systems, base types for parts of content are predefined. Content types are defined as Cartesian products from these base types. Example for such base types are: string, formatted text, integer, and content reference, indicated in Figure 6.

Web pages are filled with components. Again, a (graphical) component is not the right abstraction for pure content management since it refers to a certain layout, but we take this shortcut here. As one sample component type, Figure 6 lists a `Teaser`. Such a component shows a preview of some content, e.g., an abstract of a text, and it serves as a reference to it.

A typical content management system is furthermore parameterized with page types that name the kinds of web pages that are produced from such a system. Examples are the front page, detail pages, overview pages, search pages, etc. Figure 6 introduces an `OverviewPage` as the basis for all overview pages.

Actual web pages are defined as refinements of the `WebPage` concept or, in the case of overview pages, as refinements of the `OverviewPage` concept.

By this definition, all overview pages have a title, main content, and a news teaser. A news teaser is some teaser that displays some current information. All content parts can be refined to specific content on refinements for concrete pages. The news teaser, though, will below be used for targeting.

B. A Sample Set of Web Pages

Figure 7 shows two sample overview web pages for two parts of the web page. For sports topics, the concrete overview web page `SportsOverviewPage` is defined. Likewise, the `CulturePage` web page directs the user to the parts of the website that cover cultural topics.

C. Document Rendering

In our model, document creation (rendering) is represented by syntax rules that are attached to `WebPage`s.

If the output shall be a web page, definitions of concepts for HTML elements are needed as a basis for HTML document creation. Figure 8 shows such definitions inside the `HTML` context.

One level of concepts in the hierarchy of `WebPage` concepts can be considered a *web page template*, a blueprint for the creation of HTML documents for a range of web pages. Such a template is typically found in (website) production systems.

In our example, the above `OverviewPage` can be the template for all overview pages for the sections. Figure 8 shows a syntactical rule for `OverviewPage`.

In practical systems, typically more than one output is generated from content. In this case, not only one rendering for a web page is given. Applications are manifold: Multichannel publishing allows to output, e.g., a web page, print documents, and a mobile app from the same content. Multisite publishing allows to produce more than one website from a content base.

In order to define more than one output format for the same content in M3L, syntax rules need to be defined in contexts. Figure 8 indicates definitions of site-specific renderings in the contexts `WebSite1` and `WebSite2`.

```
SportsOverviewPage is an OverviewPage {
  Sports is the Title.
  "On this page..." is the MainContent.
}
CulturePage is an OverviewPage {
  "Museums and Exhibitions" is the Title.
  ReportOnNewExhibition is the MainContent.
}
ReportOnNewExhibition is a RichText {
  ...
}
```

Figure 7. Sample overview web pages.

```
HTML {
  <html>. </html>.
  <head>. </head>.
  <title>. </title>.
  <body>. </body>.
  ...
}
HTML {
 OverviewPage |- <html>
  <head>
    <title> Title </title>
  </head>
  <body>
    ... Title ...
    ... MainContent ...
    ... NewsTeaser ...
  </body>
  </html>. }
WebSite1 is an HTML {
  OverviewPage |- ...
}
WebSite2 is an HTML {
  OverviewPage |- ...
}
```

Figure 8. Example of a web page template.

VII. USER SEGMENTS

In order to be able to define targetable web page templates, we first need to introduce segments as predefined clusters of users with a particular interest. Segments are important for understanding user behavior and for the targeting process.

A. Base Definition for Segments

As the basis for targeting, we just define one concept `Segment` that serves as a base concept for all clusters of users. It is the means to derive user interest from user behavior, and it describes the topics of web pages.

In practice, there are other factors that influence targeting. One such factor are context parameters that take, for example, a user's device into account.

Often, also historical user data is considered. Such data may be, for example, past purchases on an eCommerce site or an explicitly maintained user profile that the user has created by registering an account for the site.

Such additional factors are not discussed here.

B. Example of Segment Definitions

Segments might be managed in a structure like shown in the example in Figure 9, assuming for our example a news site consisting of sections.

```
Segments {
  Sections {
    Politics is a Segment.
    Sports is a Segment.
  }
}
```

Figure 9. Samples segments.

Only the concrete segments **Politics** and **Sports** are significant. The contexts **Segments** and **Sections** are used to structure the set of concrete segments.

Segments are used in a twofold manner: On web page accesses, they name the topic of a web page in order to derive the area of interest of a visitor. When delivering a web page in a personalized way, a user's segment is used to select and evaluate personalization rules. These two uses will be elaborated below.

VIII. TARGETABLE WEBSITES

Since we aim at an integrated model, we take the approach of targeting that is based on personalized content selection and adaptation during document creation.

For that approach, content needs to be structured in a way that allows changing or rearranging it. On that basis, variations of content for specific user groups can be defined.

A. Base Definition for Targetable Websites

Using the M3L, personalization can be expressed in a straightforward way by providing variants of web page definitions in different contexts. Rules as used in commercial products are not required. In the case of targeting, we use the user segments as context of content variants.

This way, personalization is performed by choosing one out of different variants of a web page depending on the context of the requesting user.

As a basis for targetable web pages, we define a concept **TargetableWebPage** as a refinement of the concept **WebPage**. Like **WebPage** for conventional websites, the new base concept is not required for targeting to work. But it allows us to add some structure to our set of web page definitions. In particular, it allows distinguishing between web pages for which to apply targeting rules and for those for which not to apply them.

For all web pages, which shall be targeted, a corresponding concept is defined as a refinement. We demonstrate this using an example.

B. Targetable Website Example

To continue with the running example, we introduce a concept **TargetableOverviewPage** as the base of all overview pages in Figure 10.

As an example for a specific page, the concept **CulturePage** introduced in Figure 7 is redefined in this hierarchy. This way, it becomes subject to targeting.

The **Teaser** that we introduced as an example of a component is used for targeting in Figure 10. The example lists three such teasers to be used for users who are interested in politics or sports, respectively.

Concepts for the choice of teasers to use for specific user segments is shown below in Figure 10. Using the segments from above, these teasers can be defined as the news teaser of overview pages in the context of a specific segment.

With the definitions of this example, the news teaser referenced from overview pages evaluates to the variant for a specific segment when the page is evaluated in the context of that segment.

```
TargetableWebPage is a WebPage.

TargetableOverviewPage is a
  TargetableWebPage, an OverviewPage.
CulturePage is a TargetableOverviewPage.

LatestPollResults is a Teaser {
  ... Title ... TeaserText ... WebPage.
}
SoccerExhibition is a Teaser {
  ... Title ... TeaserText ... WebPage.
}
RunningGameScore is a Teaser {
  ... Title ... TeaserText ... WebPage.
}

Segments {
  Sections {
    Politics {
      TargetableOverviewPage {
        LatestPollResults
          is the NewsTeaser.
      }
    }
    Sports {
      TargetableOverviewPage {
        SoccerExhibition is a NewsTeaser.
        RunningGameScore is a NewsTeaser.
      }
    }
  }
}
```

Figure 10. Example of a targetable website.

Note that this holds for any overview page that allows targeting since the **NewsTeaser** refinements are made in the context of the base concept **TargetableOverviewPage**.

IX. TRACKING WEB PAGE VISITS

Web tracking is applied to websites for quite a while. Originally it was used to measure the usage of websites and it served to generate reports that allow analyzing users' behavior.

Today's tracking serves further purposes. In our case, it builds the foundation for segmentation by providing the basic usage information used to derive user behavior and, finally, user interest.

To integrate a model of tracking, we add some base definitions and add it to our running example.

A. Base Definitions for User Tracking

Figure 11 shows the base definitions **User** and **Visit** that we employ for tracking.

```
User.
Visit {
  Visitor is a User.
  ViewedPage is a WebPage.
}
```

Figure 11. Base model for tracking.

```
Anon815 is a User.
Anon815-on-SportsOverviewPage-1 is a Visit {
  Anon815 is the Visitor.
  SportsOverviewPage is the ViewedPage.
}
```

Figure 12. Example of tracking.

The `User` concept serves as the identity of a web page visitor. It may contain user data as content.

A `Visit` records the access of a user to a web page. A `Visit` thus refers to the requested page and to the user who requested it. If the user is unknown, we create a `User` concept refinement at the time of the first request. The newly created concept represents one anonymous user.

In real-world applications, typically a tracking tool is used for this purpose. In M3L implementations, `Visit` refinements are created on web page access (e.g., during HTTP request handling as discussed in Section IV as an example of a specific recognizer).

Targeting is based on the users' behavior. Behavior is analyzed by tracking web page accesses. In the example of the M3L model we do so by creating (or finding) a matching `Visit` instance for a web page and a user.

B. Example of a Web Tracking Activity

With the sample definitions made so far, assume a user is requesting the page described by `SportsOverviewPage`.

On first access of that web page, a concept `Anon815` is created as a refinement of `User` for the request. For every access of that user to that web page, a `Visit` concept for that user and the sports overview page is created as shown in Figure 12.

The suffix `-1` represents some part of the generated concept name that is used to disambiguate instances. In this example by a counter going up.

If `Anon815` next requests `CulturePage` in this example, then the system defines

```
Anon815-on-CulturePage-1 is a Visit {
  Anon815 is the Visitor.
  CulturePage is the ViewedPage.
}
```

Another request of user `Anon815` to the web page `SportsOverviewPage` would lead to the creation of an equivalent concept named `Anon815-on-SportsOverviewPage-2`. This way, every such request is tracked.

With this sample requests, the statement

```
Visit { Anon815 is the Visitor. }
```

evaluates to { `Anon815-on-SportsOverviewPage-1`, `Anon815-on-CulturePage-1`, `Anon815-on-SportsOverviewPage-2` }.

```
SegmentingWebPage is a WebPage {
  Topic is a Segment.
}
```

Figure 13. Base model for segmenting web pages.

```
SportsOverviewPage is a SegmentingWebPage {
  Sports from Sections from Segments
    is the Topic.
}
```

Figure 14. Example of a segmenting web page.

The curly brackets are not M3L syntax here, they denote a mathematical set. Since the referenced `Visit` evaluates to more than one concept, the result is a set. Further evaluation will continue element-wise.

X. USER SEGMENTATION

Based on the track records represented by `Visit` instances, the user segmentation is derived. Users shall be assigned to clusters that reflect their particular interest.

Segments can be defined in many ways. In this article, we concentrate on the analysis of user behavior as provided by tracking with respect to the segments defined for a website.

Segments are derived from visits by analyzing the topics of the visited pages. The approach is based on the assumption that the topics of the web pages that a user visits most often are the ones that the user is interested in.

A. User Segmentation Based on Interest

In order to turn track records into segment assignments this way, we add segmentation information in order to describe the web pages' meaning. More precisely, for those web pages we consider relevant for identifying the users' interest, we add the according segment to a web page definition.

To this end, we add a new concept `SegmentingWebPage` as shown in Figure 13. This concept serves as a new base concept for those web pages that are used to identify the interest of a user.

Each `SegmentingWebPage` has a `Topic` assigned. The topic is to be refined to a concrete `Segment` as defined in Section VII.

As an example for a `SegmentingWebPage`, we redefine the sports overview page in a way that helps to determine the interests of a user in Figure 14.

This way, every user requesting the sports overview page is a potential member of the `Sports` segment.

B. Determining Segments From Track Records by Scoring

Segments are assigned to users on the basis of tracked accesses to segmenting web pages. Segment information is determined in a two-step process: First, assign a relevance *score* to every user for every segment by counting accesses of that user to web pages of that segment. Second, determine to which segment(s) a user belongs. The segments that have the highest score for a particular user are assigned to that user.

The score a segment got for a user is the number of visits of a user to web pages with a topic that equals that segment. In order to compute the number of visits recorded by tracking, we need to introduce the base concept `Integer` with just enough conception in order to have the ability to count. To this end, **Integer**s have a reference **Pred** to their predecessor. Using this reference, the order of integers is defined. The numerical value of an **Integer** is thus the length of the chain of its predecessors. In M3L:

```
Integer { Pred is an Integer. }
0 is an Integer.
1 is an Integer { 0 is the Pred. }
```

On the basis of this **Integer** concept, we can define **Score** as shown in Figure 15. The **Value** of a **Score** that a segment has for a user is assigned an **Integer** concept as a refinement.

In the context of a user segment, the concept **Visit** used for tracking is redefined so that the visited pages are constrained to **SegmentingWebPage**s, and that every **Visit** created during web tracking directly leads to an update of a score.

The **Visit** directly creates a **Score** with the given user and segment through a rule. The score is set by the two refinements of **Score**.

Any score is a **ScoreUpdate**, so the value of a newly created score will be initialized with the specific **Integer 1**.

If such a score already exists with the given user and the web page's topic assigned (recognized by **Value** being an **Integer**), then it will be narrowed to the matching **ScoreIncrement**.

That concept in return will increment the value by evaluating to a **Score** with an **Integer** value that is the successor of the current value.

Else, the semantic rule will initialize the score by setting the **Value** to the **Integer 1**.

```
Score {
  SegmentedUser is a User.
  AssignedSegment is a Segment.
  Value.
}
Visit  {
  ViewedPage is a SegmentingWebPage.
} |= Score {
  Visitor is the SegmentedUser.
  Topic from ViewedPage
    is the AssignedSegment.
}
ScoreUpdate is a Score
|= Score {
  SegmentedUser is the SegmentedUser.
  AssignedSegment is the AssignedSegment.
  1 is the Value.
}
ScoreIncrement is a ScoreUpdate {
  Value is an Integer.
}
|= Score {
  SegmentedUser is the SegmentedUser.
  AssignedSegment is the AssignedSegment.
  Integer { Value is the Pred. }
    is the Value.
}
```

Figure 15. Example of targeting definitions.

```
SegmentDetermination {
  InitialThreshold is an Integer.
  SegmentsOfUser {
    UserToSegment is a User.
  }
  |= Score {
    UserToSegment is the SegmentedUser.
  }
  Score_rec is a Score {
    Value is an Integer.
  } |= Score {
    Pred from Value is the Value.
    Pred from Threshold is the Threshold.
  }
  IncludedScore is a Score_rec {
    0 is the Threshold.
  } |= AssignedSegment.
  ExcludedScore is a Score_rec {
    0 is the Value.
  } |= Segments.
}
```

Figure 16. Base model for segmentation.

On every request of a user *u* for a web page *p*, the web server issues a

```
Visit {
  u is the Visitor.
  p is the ViewedPage.
} |= Score {
  Visitor is the SegmentedUser.
  Topic from p is the AssignedSegment.
}
```

As a next step, the highest ranked segments for a user have to be found. In this article, we introduce a threshold for the rank. Segments, which have a score above that threshold, apply to a user.

The definitions in Figure 16 drive the selection process.

The highest ranked segments of a user are evaluated inside the concept **SegmentDetermination**, that serves as a scope for executions. The concept **SegmentsOfUser** acts as a function from **User**s to segments with scores above the threshold. That function is invoked within the scope.

The evaluation is based on an **InitialThreshold** that is set inside **SegmentDetermination**. It is set to the value that has to be reached by scored segments.

The first "invocation" of **SegmentsOfUser** for a user collects all **Score**s of the given user. These scores are then narrowed down during function evaluation. Each iteration of the evaluation starts through the concept **Score_rec** that sets the **Value** of the **Score** and decreases both **Value** and **Threshold** by one.

If the **Threshold** reaches **0**, then the score is narrowed down to **IncludedScore**. In that case, the value was greater than the threshold. The score is replaced with the segment in this case, thus terminating the recursion.

If the **Value** reaches **0** first, however, then the value was less than the threshold. In this case the recursion ends without a specific result by replacing the result with **Segments** as declared above.

With these definitions, the retrieval of, for example, segments of a user **Anon815** for threshold **3** is performed as follows:

```
Anon815sSegments is a SegmentDetermination {
  3 is the InitialThreshold.
  SegmentsOfUser {
    Anon815 is the UserToSegment.
  }
}
SegmentsOfUser from Anon815sSegments.
```

As an example for the application of **SegmentDetermination**, the concept **Anon815sSegments** is evaluated and projected to its content **SegmentsOfUser**, the actual "function".

With the track records from the example in Section IX.B above, **Anon815sSegments** evaluates to (equivalent to that example):

```
{ Anon815-on-SportsOverviewPage-1,
  Anon815-on-CulturePage-1,
  Anon815-on-SportsOverviewPage-2 }.
```

This set of concepts constitutes the projection of **SegmentDetermination** to **SegmentsOfUser** and, therefore, further evaluates to **Sports**: Evaluation is applied element-wise, **CulturePage** was not redefined to a **SegmentingWebpage** in our example (thus has no **Topic**), and the other two concepts both evaluate to **Sports**.

XI. Applying Targeting Rules

When users are segmented, the segmentation can be used to create personalized web pages for users.

A. *Basic Targeting Execution*

Using the model laid out in this paper, web pages are targeted by evaluating a **TargetableWebPage** in the context of the segment(s) of the requesting user. No further modeling means need to be introduced.

More precisely, the syntactical rule of a requested web page is used to produce a document for a **WebPage** concept. In the course of the application of the syntactical rule for the creation of the external representation, all concepts that are referenced in the syntactical rule are evaluated.

In case a user is assigned to more than one segment, one of the segments has to be chosen as the context of the web page evaluation. This has to be done by the environment that handles input and output recognizer and producer in the architecture (see Section IV), and is not presented here.

B. *An Example of Web Page Targeting*

With the sample concepts defined so far, the statement shown in Figure 17 targets the web page **CulturePage** to user **Anon815**.

```
AssignedSegment
 from SegmentsOfUser from Anon815sSegments {
  CulturePage.
}
```

Figure 17. An example of web page targeting.

As shown above, the segment of user **Anon815** is **Sports** in our running example. Therefore, when the page **CulturePage** is requested by that user in our example, it will be evaluated in the context **Sports**. In this context, the page will have **SoccerExhibition** and **RunningGameScore** as its **NewsTeasers**.

At the same time as the targeted web page is derived, a request for a web page may also increment the matching score as defined in the previous subsection (if it is a **SegmentingWebPage**). This concludes the circle of segmenting and targeting.

This example just demonstrates the selection of content to display at a given position in a web page, as it is also possible with commercial products. With the approach demonstrated here, however, it will also be possible to personalize other aspects of a web page in future work.

XII. Summary and Outlook

The paper concludes with a summary of this article and an outlook on future research directions.

A. *Summary*

Forms of personalization are discussed in the literature for quite some time now. Still, integrated models covering most or all aspects of personalization are missing in practice.

This paper presents a study on such an integrated model, that combines content modeling and content rendering with personalization, and that allows expressing various forms of personalization.

The initial modeling approach achieves the goal of integrating content, content representation, users, page visits, segments, user segmentation, and targeting "rules". This integration allows coherent definitions of targeted web sites.

We showed this for predefined segments and a segmentation that is derived from the topics of pages and users visiting those pages.

Technological dependencies were largely avoided. Only for tracking we rely on some additional logic that issues M3L statements based on web requests.

B. *Outlook.*

This paper concentrates on implicit personalization of presentations for groups of users, in practice called targeting.

Future research will investigate how to employ such integrated models to cover a wider range of personalization approaches and applications. With the help of such models it will be possible to use more than the set of predefined configuration options that contemporary systems exhibit. Instead, these models are expected to unveil personalization capabilities that range over all aspects of services, their content, and their appearance, as well as to give the possibility of utilizing the interconnections between these.

A first step would be to extend the model to other forms of personalization in order to investigate whether these fit in equally well and can be combined within one model.

Content delivery and consumption depend on the context of the user. The utilization of context information for personalization should fit the models well using the M3L. Still, this needs to be studied.

This paper covers an analysis based on a hypothetical model only. It now needs to be connected to a working web server in order to gain practical results.

To increase practical relevance, further information on users should be integrated into the targeting process. Besides the segments derived from user behavior, additional parameters of the context of the user can be taken into consideration.

Contemporary systems incorporate base data of website visitors that have explicitly been raised using other channels, e.g., previous visits or interactions on other channels. Such information may come from a Customer Relationship Management (CRM) system, from transaction processing systems like shop solutions, and from customer journeys.

In practice, the whole information on users may be centralized in a Customer Data Platform (CDP) that contains both the explicit data from the CRM as well as implicit information coming from tracking and segmentation. With such a CDP, targeting is applied to a higher degree than the personalization of single web pages, e.g., by contributing to omnichannel orchestration and customer journey orchestration.

For the examples in this article, it is assumed that a web server is incorporated into systems, and that it can be extended in a way that it analyzes HTTP requests and translates them to some M3L statements. For an even more coherent integrated model of targeting, and in order to avoid the need of additional mapping code outside the models, the HTTP protocol should be modeled in the M3L. E.g., a syntactical rule

```
WebPage |- "GET /" the name "HTTP/1.1".
```

(assuming the concepts with the names in quotes are defined) maps HTTP requests to concepts. Instead of just producing HTML as shown above, further syntactical rules can produce full HTTP responses. This extension of the integrated model has not yet been done.

ACKNOWLEDGMENT

Targeting is one building block in many digitization projects. Valuable insights have, therefore, been gained thanks to exchange with colleagues, partners, and customers.

Though the model presented in this paper is not directly related to the concrete project work at Namics, the author is thankful to his employer for letting him follow his research ambitions based on experience made in customer projects.

REFERENCES

[1] H.-W. Sehring, “An Integrated Model for Content Management, Presentation, and Targeting,” Proc. Eleventh International Conference on Creative Content Technologies, May 2019, pp. 26-31.

[2] K. Riemer and C. Totz, “The Many Faces of Personalization: An integrative economic overview of mass customization and personalization,” in The Customer Centric Enterprise, M. M. Tseng and F. T. Piller, Eds. Berlin, Heidelberg: Springer, pp. 35-50, 2003.

[3] J. Blom, “Personalization: a Taxonomy,” Proc. CHI EA '00 Extended Abstracts on Human Factors in Computing Systems, Apr. 2000, pp. 313-314.

[4] J. Zhang, “The Perils of Behavior-Based Personalization,” Marketing Science, vol. 30, pp. 170-186, Dec. 2011.

[5] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen, “How much can behavioral targeting help online advertising?” Proc. 18th International Conference on World Wide Web (WWW '09), April 2009, pp. 261-270.

[6] A. S. Das, M. Datar, A. Garg, and S. Rajaram, “Google news personalization: scalable online collaborative filtering,” Proc. 16th International Conference on World Wide Web (WWW '07), May 2007, pp. 271-280.

[7] Y. Chen, D. Pavlov, and J. F. Canny, “Large-scale behavioral targeting,” Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09), June/July 2009, pp. 209-218.

[8] A. L. Montgomery and M. D. Smith, “Prospects for Personalization on the Internet,” Journal of Interactive Marketing, vol. 23, pp. 130-137, Jul. 2008.

[9] H.-W. Sehring, “Content Modeling Based on Concepts in Contexts,” Proc. Third International Conference on Creative Content Technologies, Sep. 2011, pp. 18-23.

[10] J. W. Schmidt and H.-W. Sehring, “Conceptual Content Modeling and Management,” Perspectives of System Informatics, Lecture Notes in Computer Science, vol. 2890, pp. 469-493, 2003.

[11] H. Beales, “The Value of Behavioral Targeting,” Mar. 2010. [Online]. Available from: https://networkadvertising.org/pdfs/Beales_NAI_Study.pdf [retrieved: December, 2019]

[12] P. T. Aquino Junior and L. V. L. Filgueiras, “User modeling with personas,” Proc. 2005 Latin American Conference on Human-Computer Interaction (CLIHC '05), Oct. 2005, pp. 277-282.

[13] T. Jiang and A. Tuzhilin, “Improving Personalization Solutions through Optimal Segmentation of Customer Bases,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, pp. 305-320, Mar. 2009.

[14] J. R. Mayer and J. C. Mitchell, “Third-Party Web Tracking: Policy and Technology,” Proc. 2012 IEEE Symposium on Security and Privacy, May 2012, pp. 413-427.

[15] K. S. Kuppusamy and G. Aghila, “A Model for Web Page Usage Mining Based on Segmentation,” International Journal of Computer Science and Information Technologies, vol. 2, pp. 1144-1148, 2011.

[16] Website of the CoreMedia AG. [Online]. Available from: https://www.coremedia.com/ [retrieved: December, 2019]

[17] .NET Daily. *How to Use Adaptive Personalization with Sitecore* [Online]. Available from: https://dotnetdaily.net/sitecore/how-to-use-adaptive-personalization-sitecore [retrieved: December, 2019]

[18] Adobe Target. *Behavioral targeting*. [Online]. Available from: https://www.adobe.com/marketing/target/behavioral-targeting.html [retrieved: December, 2019]

[19] M. Gorgoglione, C. Palmisano, and A. Tuzhilin, “Personalization in Context: Does Context Matter When Building Personalized Customer Models?” Proc. 6th IEEE International Conference on Data Mining (ICDM 2006), Dec. 2006, pp. 222-231.

[20] H.-W. Sehring, “Context-aware Storage and Retrieval of Digital Content: Database Model and Schema Considerations for Content Persistence,” International Journal on Advances in Software, vol. 11, pp. 311-322, Dec. 2018.

[21] T. J. Parr, R. W. Quong, “ANTLR: A predicated-LL(k) parser generator,” Software-Practice and Experience, vol. 25, pp. 789–810, Jan. 1995.

[22] T. J. Parr, “Enforcing Strict Model-View Separation in Template Engines,” Proc. 13th International Conference on World Wide Web (WWW '04), May 2004, pp. 224-233.