

Content Design and User Interface Architecture for Multiscreen Projects

7

Methods and Rules for the Conception, Design, and Implementation of Layout, Contents, and Workflows in the Building Block Principle

Digital contents can appear everywhere today. It is just a fact that we use digital services daily on a wide array of devices and media. Information flows into all channels. In order to create a uniform user experience, a continuous flow of information is required. The requirement for this is a central hub for contents (as described in [Chapter 6](#)), a system for the definition of user interface (UI) elements, and rules regarding when which contents are displayed in which combination where and how. So that this can be solved technically, it is necessary to plan and develop the contents, user interfaces, and workflows based on a respectively similar model in a modular and structured manner—comparable with the building blocks in a building block system. Based on these requirements, there is a distinct need for design systems (Busse, 2015). This chapter describes how to combine well-known and established methods and patterns from content strategy, information architecture, content modeling, and UI design to create relationships in order to develop new added-value approaches.

Information distribution is changing disruptively. Media publishers and content owners (which is nearly everyone that offers any kind of information on the web) are no longer in control of how and where their contents are distributed and retrieved (see [Chapter 6, Section 6.1](#)). But they can prepare their content (structure, flow and appearance) for such a scenario.

7.1 A CRUCIAL SHIFT FOR CONTENT (STRIKING NEW PATHS)

Many content offerings (e.g., individual articles from *The New York Times* or *Der Spiegel*) are today more frequently available in other channels (e.g., on Twitter, Facebook, WhatsApp, in an e-mail, in Flipboard or any other third party apps) than the original website. A content interface (content application programming interface, content API) supplies all information, and the recipient or the displaying channel decides which content elements to use. The visual design is then prescribed by the styling definitions of the target and third-party platforms.

own channels

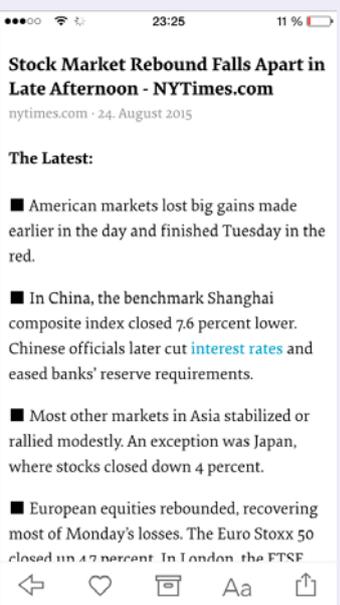


website, smartphone

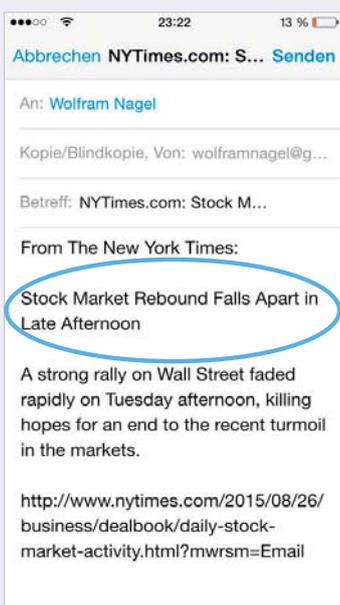


iPhone app

third party channels



Instapaper



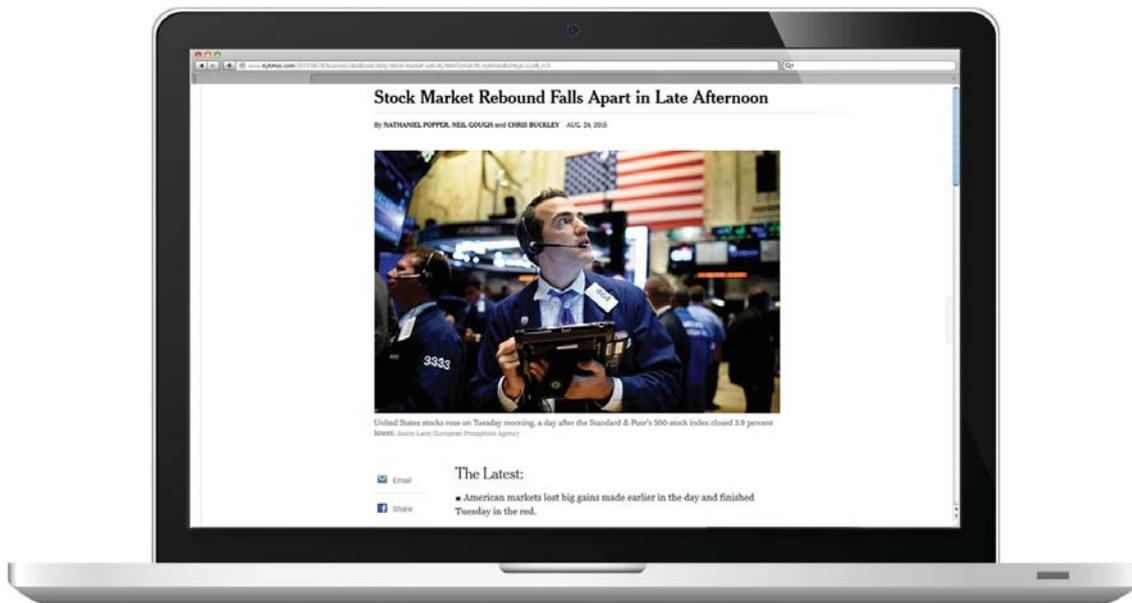
2nd row: Sharing content from the smartphone browser via e-mail, on Facebook (incl. timeline) and on Twitter (incl. detailed view with source summary)







Displaying content from the same source and origin on different devices, for example smartphone app, smartwatch (push notification), and the website on the laptop or desktop-PC.



The content of news companies (for example the New York Times) can appear in many different channels and devices. These can be (on) the **own channels** (respectively touchpoints) as the original website in the browser, on the tablet or in the smartphone app, but **third party channels** as well, that can't be controlled by New York Times editors and designers. The content and its structure has to be appropriate for the original website on desktop or smartphone, for own or third party apps (like Instapaper), E-Mail Sharing, Facebook, Twitter (with or without summary), WhatsApp news channels, smartwatch or smartphone notification, and many more. **In some of the examples you see the marked headline of a news used in different use cases (e-mail sharing, smartwatch notification, twitter post).** Some content owners also have to publish their content on TV as well (just one more device with different requirements for content structure, complexity, and presentation).

In order to be able to supply the right content building blocks for automated usage, the content must be correspondingly structured—in the best case, even “intelligently.”

You can create good experiences without knowing the content. What you can't do is create good experiences without knowing your content structure. What is your content MADE from, not what your content IS.

Mark Boulton

7.1.1 MULTISCREEN-READY INFORMATION EXPERIENCE

The Internet is an integral part of daily life; digital information is used on a wide array of devices. We live in a multiscreen world. The art and manner of how information is created are changing. On the side of the creation of the content is the author or curator of the content; on the other side is the recipient and thus the user of the information.

An information management system must support the work of content creators and should be able to serve as a hub for all contents. Its interfaces (API), data aggregation, and workflows must be able to be designed individually.

The collection and/or creation of the content seldom occurs via a uniform system. It would be best if authors could use their customary creation system and the collected contents could be consolidated, merged, and brought to the defined structure via corresponding (application programming) interfaces and a kind of content structure mapping in the content hub based on the content model. By so doing, it would be possible to collect contents based on and according to the structure of the content model and the content type with any system and via any input-channel (e.g., via e-mail, Evernote, a Tweet, a Facebook post, or another system), to edit it in the content hub, and to flexibly transmit and publish it again to any desired channel.

Content structure mapping (in a centralized content hub/exemplified)

	E-Mail	Evernote	Twitter	Facebook
Title	subject	title of the notice	–	–
Short text	–	–	tweet text	text of post
Long text*	mailtext	notice text	linked content	linked content
Image	attachment	integrated	attached	attached
Document	attachment	reference	linked file	linked file
Date	mailing date	last date of change	date of tweet	date of post
Author	sender	originator	account	account

Content elements can be generated via variable input channels (each channel quasi serves as CMS) and then be mapped according to the underlying structure of the content type (for example a news article, compare New York Times example).

**Semantic structuring (e.g. H1, bold, quote, listing, etc.) is adopted.*

Thus, it would be possible—via a “simple” e-mail—to collect the content elements (of a news article, for example) of author/creator, creation date, title, brief description (short text), main text contents (long text), photos or media elements (as attachment), and other meta-information, for example via hashtags or other defined characters in the e-mail text as the Todo app Todoist does (www.todoist.com). The gathered contents could then be edited, optimized, and supplemented in the central hub.

Content providers are confronted with the problem that they should be able to consistently publish contents in many channels. Information should be able to be displayed everywhere. The requirements for this are a central and, insofar as this is possible, user-friendly hub for contents (e.g., content management system [CMS]), a central system for the definition of UI elements (e.g., style guide or UI pattern library), and corresponding rules (workflows) regarding when which contents will be displayed in which combination on which device and how.

In an upstreamed or at least a content manager experience design (CMXD) process initiated at the beginning of the project, the most important parameters for this can be defined (Nagel, 2014b) (see Chapter 6, Section 6.4.3). This includes recognizing the modes of content collection, defining the three to five most important target channels, and correspondingly adjusting or synchronizing the input interface and the systems (see Chapter 6, Section 6.3.3).

7.1.2 AN INDEPENDENT METHOD FOR AN UNCERTAIN FUTURE

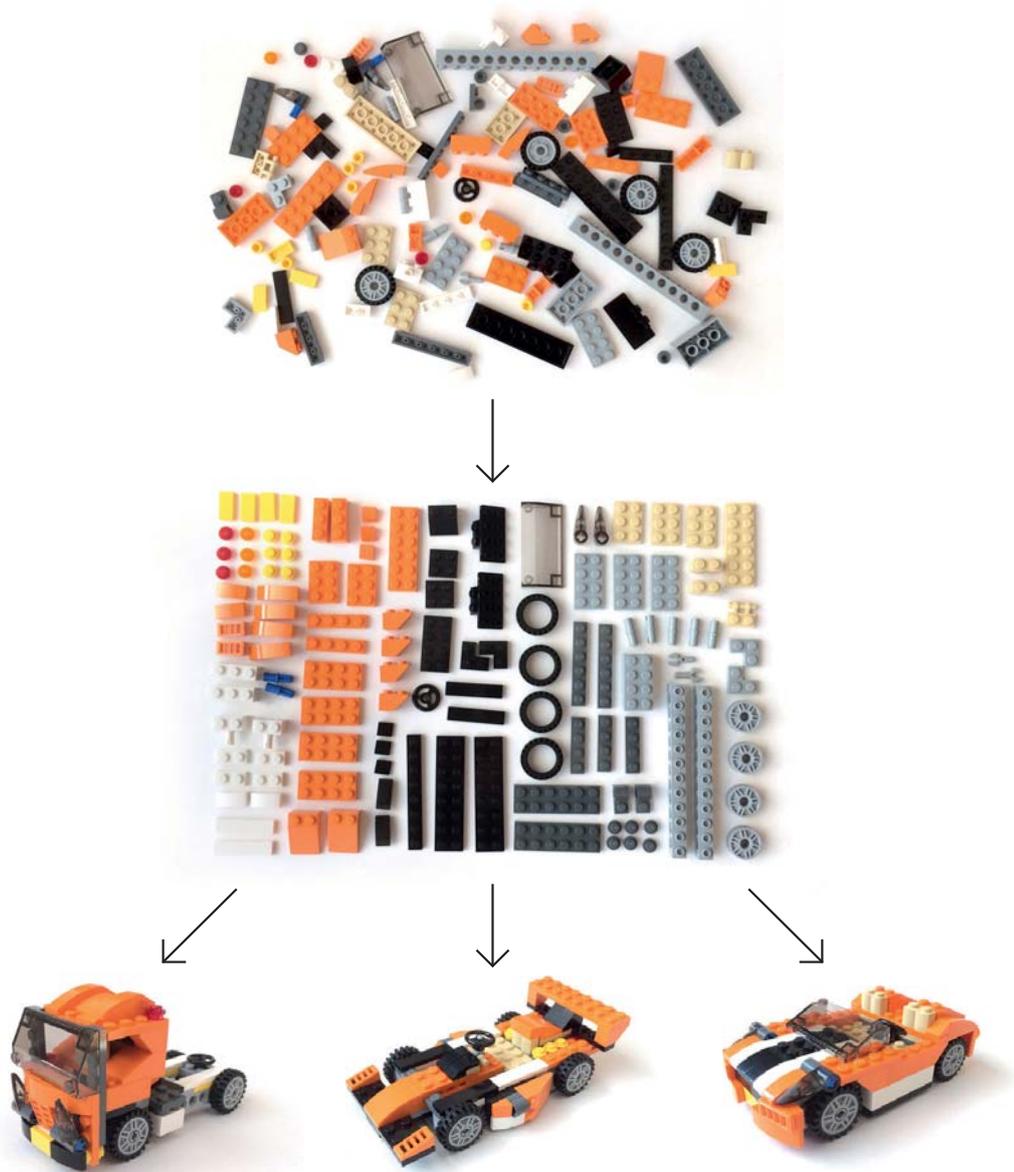
It concerns, among other things, the fundamental principle and a uniform generally understandable language and interdisciplinary terms that are understood by all stakeholders (project managers, customers, developers, conceptualists, and designers). The core idea is to break down everything (similarly) atomically and/or in accordance with the building block principle and/or compile it and generically conceive it. Most important, we cannot anticipate for which future channels we will be designing contents and user interfaces at the current point in time (Frost, 2015b).

I have no idea what the hell I am doing. And neither do you. And that's ok.

Brad Frost

I think that if you create contents, user interfaces, and processes based on atomic, modular building block patterns, you will be largely independent of future developments.

Semantically similar contents should be variably utilizable in various media, channels, and devices. This applies to the smartphone, tablet, desktop, TV, and smartwatch as well as for contents on your own website, in Google search results, in the RSS feed, on Instagram or Facebook, or any other content presentation.



If you have a bunch of unsorted LEGO bricks (comparable to a lot of content and UI elements on a website, for example) you can (and should) first sort them to get an overview. **Make an inventory** of all the bricks that are available and shall be used to build any LEGO model. You can build different cars out of the same elements, if you combine them in a different manner. That's the same what you can do and must think about with your content (and your user interface). Different elements can and will be shown and differently used and combined on various channels and touchpoints (compare example with New York Times in [Section 7.1](#)). A systematic and structured approach helps to tackle that challenge.

7.3 PROCESSES AND TOOLS

If you select the atomic approach and gradually begin with the smallest elements that build on one another—the atoms—you will receive presentable pages only late during the course of the project. The requirement for this is that mood boards, scribbles, or Photoshop (mood) screens that indicate the visual design direction and, in my opinion, still have legitimacy, are sufficient for coordination purposes at the beginning. The more the project progresses, the more the initially higher expenditures at the project start pay off (Schätzle, 2015b).

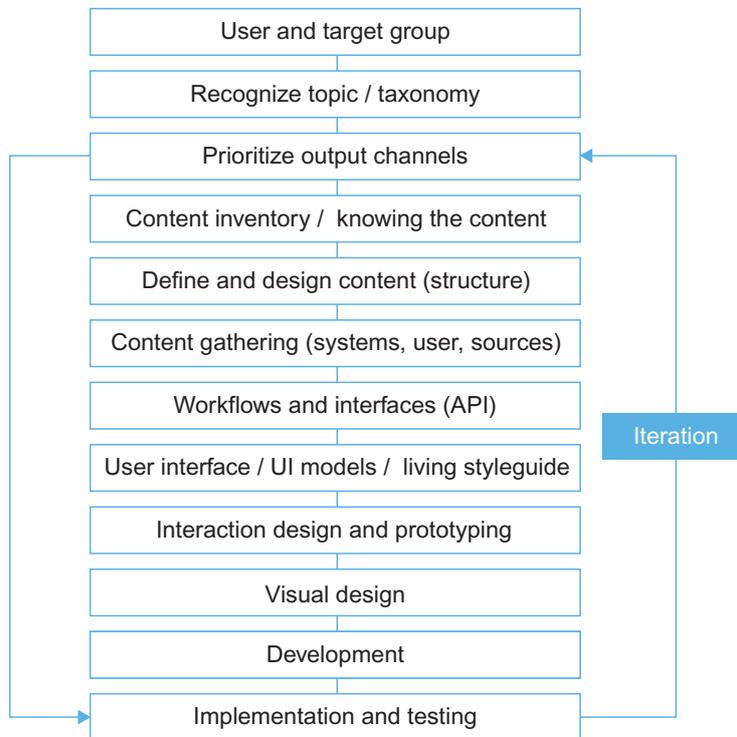
A project from a user experience perspective is broken down basically into the disciplines of user research, content/information architecture, user interface, interaction design, visual design, and development/implementation.

During the course of a project, various methods should be employed that can be allocated across these disciplines. Due to budget, competence, time constraints, or other reasons, never all relevant methods can be applied to a project. However, you should at least focus on them and be clear about their relevance and benefit for the overall results. With the right method at the right time, unnecessary resource consumption during the course of the project can be avoided because you have to “stumble around” less and decisions can be made and justified based on solid findings.

7.3.1 METHODOICAL APPROACH

The disciplines of UI modeling and content modeling, visual design, and multiscreen conception will overlap even more in the future. Projects will become more complex, and all disciplines and thematic areas will have to be considered and handled at the right time.

During the course of the project, you should (compactly formulated)—sequentially and iterated—recognize the theme, prioritize output channels, and define the content models. Based on the content, additional details can be defined and handled (structure, content wireframes, process of content collection, and application programming interfaces). When these parameters are known, while taking the building block principle into consideration, workflows can be modeled and output interfaces identified, designed, and developed. Such a process is seldom linear. Project phases can overlap, change and swap. Because content is most important, a central content (management) hub could address the previously mentioned plan. A more detailed example of a multiscreen project workflow with further information and recommended methods can be found at <http://www.msxbook.com/en/pjwflw>.



Project workflow: Start with the users and the topic. When you know these initial points you can successively get to know and work on the content. The user is always important, that means that authors, editors and recipients should always be taken into consideration. Don't forget to consider the potential context of use. Content modeling, UI modeling, visual design and development are iterative and **overlapping project phases. Such a process is seldom linear, phases can parallel, change and swap.** When you think you're done, think again (if possible). Define, create, refine, repeat...

This is just a rough overview. There are a lot of helpful methods, that have to be considered and applied (a few are mentioned in the previous chapters).

7.3.2 CENTRAL CONTENT (MANAGEMENT) HUB

The idea and the concept of the hub are based on the aforementioned breakdown of the four project parts of content, user interfaces (UI), workflows, and application programming interfaces (API). The content editor (thus, the user of the software) is in the foreground. The data structure for the back end is defined based on the structure of the content models and/or types. The editor interface consists of a form based on the data structure. In the central content hub, the contents can be filed and managed in a consolidated manner.

The content model defines the UI model

Even the back end has a front end. If a content audit is conducted at the beginning of a project and then the content models are created, the suitable UI form elements can be derived for the back end (and the UI elements for the front end). Through the correlation of content and UI, the content model fundamentally structurally also already defines the back end and the output in the front end.

Thematic relevance

The (project) theme is relevant with regard to the terminology for the content element and the labeling on the back end and/or the user interface for the collection and management of contents (unit, domain, and taxonomy). It makes a difference whether it is about cooking recipes, technical products, restaurants, business address data, or a request for vacation leave. The language and accordingly also the name of the labels for the content elements in the content collection system (classically a CMS back end) will be different.

Based on the respective unit- or theme-specific content models, a kind of specialized content manager will be derived that is suitable for the defined use case. For each defined content type, the user will receive the correlating UI form elements. A content model for a cookbook with recipes is substantially different from the content models for a product catalog with an e-shop functionality, a calendar application, or a classical news publication.

Via a form generator, the form elements can be generically combined according to the building block principle for the input interface. Through the content type, the predefined structure of the form type is fixed (atomic correlation of content, back end UI, and front end UI) but could nonetheless be expanded as required with own modules and templates.

Independent input interface

The data migration into the content hub can in principle occur independently of the input interface. The content objects must merely be correspondingly converted and its atomized elements be allocated to the particular content elements (content mapping).

The fundamental (structure of the) UI for the input interface is defined through the structural definition of the content types. But in contrast the exact styling is not yet defined (that is flexible and structure-independent). In addition, own UI elements can be supplemented at any time. The output is likewise independent of this and can take

place in an own channel or in a third-party channel. The styling of the content elements in the front end is independent from the back end. Merely the structure (chunks of the content objects, not their order) is prescribed. The view mode in the back end is “only” a “content structure preview.” The order, arrangement, and display of the content elements occur in the client and/or output medium.

Through the clear structuring and classification of the content types (concretely filled content types are content objects), the data are basically platform- and touch-point-independent and can be displayed in as many output media and channels as desired. The output is configured on the back end (via rules regarding the respective contexts of use that again contain standards regarding channels, the layout and the medium, and also the interaction). For each content element, one or more UI forms must be available for your own channels.

Content flow

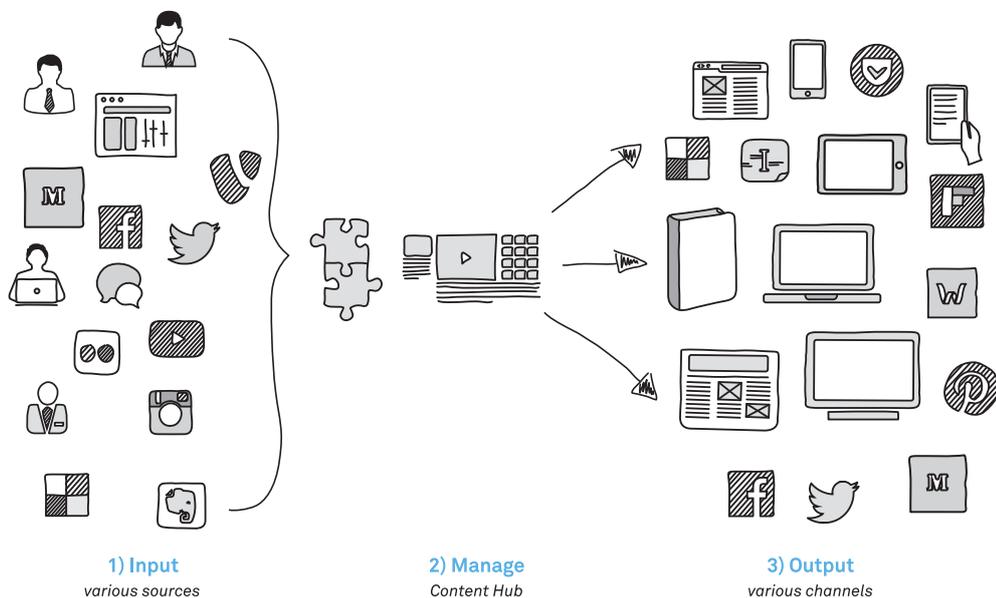
There are three parts in a content’s (work) flow. It has to be created (written and structured). It has to be managed (by adding meta data, defining workflows and rules, and setting it up). And it has to be delivered to any kind of target channel or viewport where it is viewed and received by the (end) user. It could also be a preview check how it may roughly look anywhere.

The challenge for a content management software is to deal with these three steps equally well. If one solution cannot fulfill these requirements sufficiently, perhaps you can solve that challenge by using and combining different solutions via (application programming) interfaces (Boulton, 2013).

Three-step content hub

The content hub is structured in three steps (see figure in Chapter 6, Section 6.3.3 and the following figure). You can create, collect, and output contents—wherever and with whatever you would like.

For the output, a corresponding interface (API) is required. Thus, for example, you can input and collect content via the input form and/or in the CMS user interface of the content hub and then output it in WordPress, Flipboard, or any other channel). It is also conceivable to collect contents in any system as desired (step 1, flexible input interface), and after the mapping of the content in the content hub (step 2), they are once again flexibly outputted (step 3).



The **Three-Step Content Hub** (principle): 1) Collecting and authoring 2) combining and managing 3) outputting and distributing to various channels. The content hub in the middle plays a central and important role in the content flow. Content will be aggregated, mapped to a predefined structure (if there are different input channels), organized and prepared for flexible output in the content hub (see also the content mapping example at the beginning of this chapter).

Combining theory and practice

It is hoped that a tool will be developed that can be used to combine processes and methods from content modeling and UI architecture because it then serves as an interface (or link) between these both disciplines. It would unite the theory and the methods with the practice and provide concrete benefits for the project.

Whenever I can implement content modeling directly in or with the content (management) hub, I quasi automatically receive (generic, automated, but individualizable) work results that I can use for the interface and depiction for the recipient. One has to do everything only once, and directly gets a concrete result.

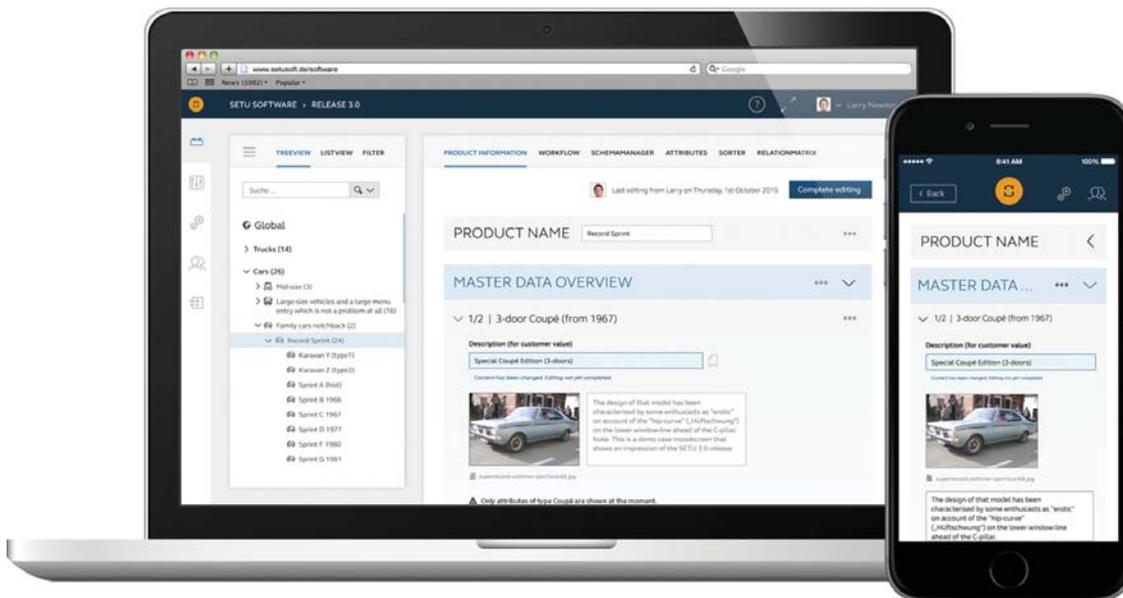
7.3.3 A MODELING SOFTWARE THAT IS SUITABLE FOR THE BUILDING BLOCK PRINCIPLE

SETU as the tool for future-oriented content

Based on the aforementioned ideas, we at SETU GmbH (www.setusoft.de) have developed a concept for such a “content management hub” that is integrated into our SETU 3.0 release. With it content management apps can be realized for various thematic focuses (units).

The content architect has predominantly free rein with the structuring of the data bodies. The software offers a large library of predefined (and extendable) form and output elements and supports iterating elements within a data model (including more complex structures such as text with an image) as well as relationships and kinds of relationships to other elements. It offers functions for the creation and management of language- and market-specific contents such as heredity and workflow support. Existing platforms can be supplemented with freely definable workflows in customer communication.

Moreover, rule-based and connected and related and individually personalizable context-based contents can be supplied. Context relevance can be recognized and utilized via rules and meta-information. Personalized contents dependent on, among others, the log-in, user profile and user type, Internet speed, device, output medium, channel, etc. (see Chapter 4 for more details in this regard). The structured information that has been supplied can be integrated into as many platforms and channels as desired.



SETU 3.0: **Content Modeling Software** suitable for the building block principle. The mood-screen shows an impression of the user interface of a demo use case of the SETU 3.0 release (www.setusoft.de).

7.3.4 QUO VADIS CONTENT AND USER INTERFACE?

For the future, it is recommended to plan and collect information in any shape according to the building block principle in order to publish information in all conceivable channels with the right tools and without large additional expenditures insofar as this is possible. The suitable principles, methods, and processes should be described,

established, and applied in a comprehensible manner. Out of this conviction, insofar as this is possible, I recommend that you break down all project parts into generic individual parts that can later be easily assembled once again—quite similar to how you can do this with LEGO.

We live in a multiscreen world. The number of screens and objects that have to receive, process, depict, and display content in any imaginable form is growing. Thus, it is highly recommended to have one single point of truth (for content and UI)—a centralized content hub with a user-friendly interface and an appealing editor-centered user experience (see Chapter 6, Section 6.4) and a living style guide and/or a UI pattern library (see Chapter 5, Section 5.8.1) to manage that challenge now and in the future.

7.4 CONCLUSION AND TIPS

1. The (digital) content appears everywhere!

We live in a multiscreen world and use digital services daily on a wide array of screens. Information flows into all channels. Content offerings have to be available on own and foreign third party touchpoints.

2. Know your content.

Conduct a content audit and/or content inventory. Analyze your content (structure) to start designing based on these findings.

3. Prepare consistent and everywhere publishable content.

Only when content is collected, stored, and edited in one place, you can avoid redundancy, guarantee consistency and are able to structure and prepare it for the publication to and the usage in every imaginable context.

4. Think about four core areas.

(Beside the user, context, and screens) Content, user interfaces, workflows, and APIs are the basic requirement to be able to publish content on a huge variety of screens and diverse channels. Start with the content.

5. Flexibility by using the (generic five-stage) building block principle.

Strip content and user interfaces down to the smallest possible elements. Built content and UI in a modular and structured manner based on these elements. Define and built larger components and concrete objects as you would do this with LEGO® bricks.

6. The content model defines the UI (model).

Start with the content and use the correlation between content elements and UI elements (in the back end and the front end) to design your user interfaces and plan (for the uncertain) appearance of your content in different channels and touchpoints.