6. COMPARING OUR PROPOSAL

6.1 Comparison Criteria

In order to compare and discuss the main characteristics of the different approaches, we developed an evaluation framework, as Figure 6.1 shows, which is divided into three main criteria: *Accessibility*, *Design* and *Other criteria*. Each of these topics deals with different issues of the approaches in order to describe them and analyze their strengths and weaknesses when developing an accessible Web site and from a Web engineering perspective. Following, we explain the meaning of the three main criteria through their set of topics.

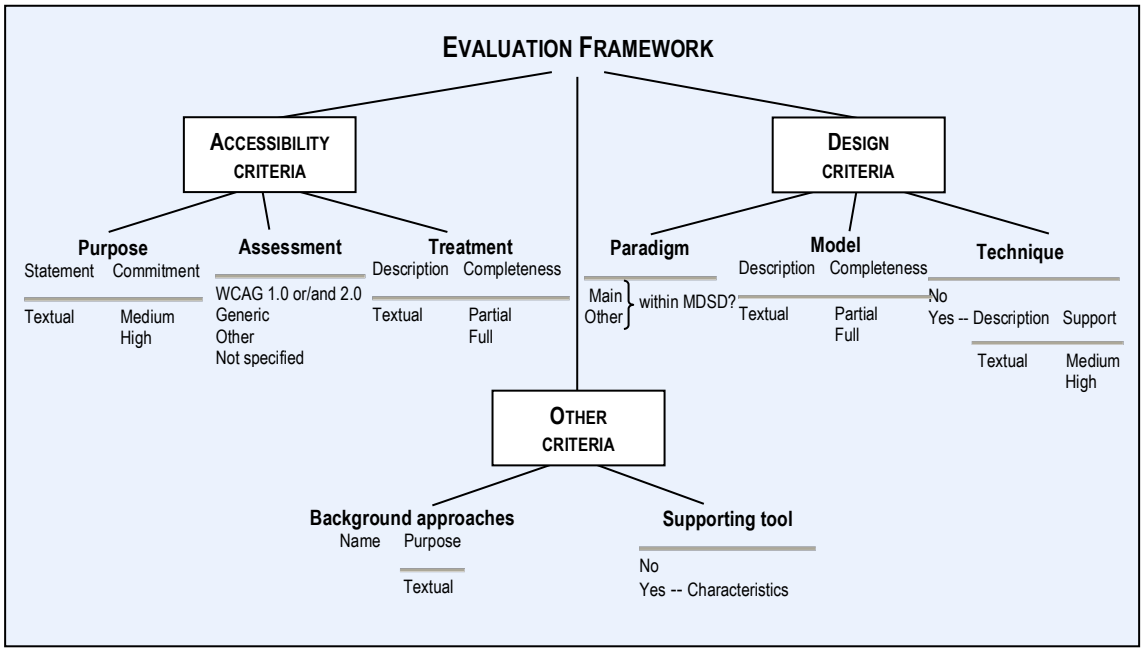

Figure 6.1: Evaluation Framework

Accessibility criteria. We propose these criteria to assess the degree of commitment with Accessibility by evaluating three topics: *purpose*, *assessment* and *treatment*.

We analyze the *purpose* earliest and in the context of the *Accessibility criterion*, because the main focus of our evaluation is on the support given to Accessibility during a Web site development process. Here we evaluate the degree of commitment to Accessibility by considering only two possible scores --i.e. "medium" and "high", because we have already selected approaches with a certain relation with Web

Accessibility. So a “low” score is out of range for the purpose of this comparison. The differences between the “medium” and “high” scores are set depending on whether Accessibility is the main concern of the approach under consideration.

In addition, because the results can be broadly different depending on the applied reference guidelines, the *assessment* topic aims to establishing the Accessibility conformance criteria applied by the approach. In this case the options are “WCAG” (1.0 or/and 2.0)[50] [48][49], “generic”, “other” or “not specified”. We are particularly interested on those approaches applying WCAG guidelines because as we said before it is a World-Wide reference normative. We choose “generic” when the approach proposes to consider standards and guidelines develop for several domains[51], such as Accessibility for e-Learning, software, PDF format, Java language, media and Web content, but it does not apply directly to any particularly. An “other” choice states that the approach can apply any “other” practice --e.g. using an ontology, an heuristic, a markup framework, etc., to analyze and treat Web page Accessibility at some stages of the development process --e.g. analysis and design, implementation, etc., and to generate an accessible Web page version. Finally, we decided to include a "not specified" choice for those approaches whose focus is not exclusively on Accessibility, so they do not need to model using a particular Accessibility principle, standard or guideline.

Finally, the *treatment* topic refers to the way Accessibility is handled by the approach. In addition it is important to highlight that many other issues can be taken into account related to Web Accessibility requirements, for example, the type of user disability --i.e. visual, motor, cognitive, deaf, etc. For the *treatment* topic, we are particularly interested in establishing how the approach deals with Accessibility requirements during a Web site development. We believe that Accessibility should be considered as part of the Web design process instead of being evaluated by a post-design repair process. This is the reason why at the analysis of this topic we are mainly interested on establishing the degree of completeness with which the approach handles Accessibility through the stages of the development process. For the purpose of evaluating the *treatment* topic we

[50] An Overview to WCAG Standards at http://www.w3.org/WAI/intro/wcag.php

[51] A list of Accessibility resources at http://www.accesstechnologiesgroup.com/Resources

provide a brief description to highlight the stage (or stages) of the design process where the approach concentrates the Accessibility efforts. Then we evaluate the degree of completeness using only two possible scores --i.e. "partial" and "full", because we selected approaches with a certain relation with modeling Accessibility. So a "low" score is out of range for the purpose of this comparison. We set a "full" score when the approach allows the integration of Accessibility from an early stage, and gives support through the whole Web design process; otherwise, a "partial" score is set.

Design criteria. We propose these criteria to evaluate design issues of the approaches under consideration by using three topics: *paradigm*, *model* and *techniques*.

At the *paradigm* topic, firstly we are interested in identifying if a main paradigm or some other combination of paradigms is used by the approach to deal with Accessibility at design. Since our comparison is framed within Web Engineering (WE) principles, we are also interested in identifying if the approach follows a Model-Driven Software Development (MDSD)[52] as the core operational *paradigm* to drive the development process. This kind of approaches are usually classified as Model-Driven Web Engineering (MDWE) [31], since they address the different concerns involved in the design and development of a Web application using separate models (such as content, navigation and presentation), and these models can then be supported by model compilers that produce most of the application's Web pages and logic right from the original models [31]. In consequence, we propose "main", "other" or "main/other within MDSD" options for the *paradigm* topic. At this point it is important to highlight that we are specially focusing on approaches using the AOSD paradigm to deal with Accessibility at design, because we believe that aspect orientation allows managing Accessibility's nature properly and as a first-class citizen.

The *model* topic refers to models provided by the approach to deal with Accessibility, and in particular the user interface *model,* since it is at the user's interface level where

[52] As we already said, one of the best-known MDSD initiatives is called Model-Driven Architecture (MDA) from OMG at http://www.omg.org/mda/One. The MDA framework, together with its related acronym Model-Driven Development (MDD), are registered trademark of the OMG, trademarks within the Unified Modeling Language (UML) is central. Web Engineering is a specific domain in which MDSD can be successfully applied.

Accessibility barriers mostly shown. We introduce in first place a brief description of the basis of the *model* proposed by the approach. It is highly desirable that this *model* fully maps the criteria assumed for treating Accessibility --i.e. the *treatment* and *model* topics must be in concordance and reinforce each other. For the purpose of the *model* topic evaluation, we focus on what elements of an interface model are addressed by the approach and how they are addressed taking into account the fact that these elements are the media for holding an Accessible user-system interaction. We suggest two possible scores, "partial" and "full", to define the degree of completeness with which the *model* specifies the interface elements. We propose to analyze this degree of *model* completeness from three perspectives: (i) the quantity and granularity of the interface elements considered by the *model*; (ii) the level of detail with which the *model* represents these elements; and further, (iii) the consistency and continuity of a main paradigm with which the approach defines and applies the *model* to deal with the Accessibility of the interface elements. We attach a "full" score, when the model provides the necessary mechanisms for dealing with the Accessibility required by the interface elements. Otherwise, we set a "partial" score. Again, a "low" score is out of range because of the selected approaches for the purpose of the comparison.

Finally, we introduce the *technique* topic to consider the case in which the approach proposes some proprietary technique to complement itself. In the case of an affirmative answer, we provide a brief description of the *technique* and its name --if any, and we also evaluate this *technique* from the perspective of providing support to enrich the design level and to reinforce the Accessibility treatment. When the technique is specifically proposed to provide this kind of support we score it as "high"; otherwise we use a "medium" score.

Other criteria. We propose these criteria to consider two additional topics: *background* and *supporting tool.* We include the *background* topic to consider the case in which the approach takes into account and/or is based-on previous work. Since we believe that the approach's basis is relevant to the approach's strength, for each previous work we provide the name and the purpose within its respective approach.

Finally, we introduce the *supporting tool* topic to indicate whether the approach has an associated supporting tool or not. Also it is important the kind of support given and

features covered by the tool in order to contribute to the development of an accessible Web application. Therefore, if the approach provides a tool, some extra considerations about the characteristics of the tool are also given here.

Table 6.1: Accessibility Criteria applied to the six approaches

ACCESSIBILITY CRITERIA					
Approach	**Purpose**		**Assessment**	**Treatment**	
	Statement	Commitment		Description	Completeness
A1 **Plessers et al.** [35]	Generate the semantic annotations (authoring and mobility Accessibility concepts) for visually impaired users as a by-product of the Web design process.	High	Other	Applies its own developed semantic annotations through a transformation process at the WSDM Implementation Design phase.	Full
A2 **Centeno et al.** [9]	Provide Accessibility support in a Web composition process managed by a design tool.	High	WCAG (1.0)	Uses a set of compliance rules, which are based on the WCAG 1.0 checkpoints, to ensure accessible Web pages from the composition of accessible HTML snippets.	Partial
A3 **Casteleyn et al.** [6][7][8]	Engineering Adaptation concerns to extend an existing HERA-based [23] Web application.	Medium	Not specified	Applies aspect-oriented techniques to add Adaptation concerns in a high-level specification and separate from the regular Web process.	Partial
A4 **Zimmermann & Vanderheiden** [53]	Introduce a process model for Accessibility design that includes well-known software engineering tools.	High	Generic	Develops Personas to support Accessibility requirements and links them to Accessibility guidelines and checkpoints for conformance testing.	Full
A5 **Moreno et al.** [29][30]	Introduce AWA module that is a domain-specific metamodel of the Web Accessibility domain.	High	WCAG (1.0) (2.0)	Identifies meta-objects following the standard WCAG.	Full
Ours **Martin et al.**	Early engineering of Accessibility concerns within a Web development process.	High	WCAG (1.0) (2.0)	Models Accessibility as an aspect-oriented concern moving from abstract to concrete architectural views.	Full

6.2 Discussion

At this point we are ready to evaluate the six approaches in accordance with the characteristics defined by our evaluation framework. To make more understandable our explanation, we refer to the approaches as A1 [35], A2 [9], A3 [6][7][8], A4 [53], A5 [29][30] and Ours.

Accessibility criteria. Table 6.1 shows the resultant evaluation of the Accessibility criteria applied to the six approaches. As we can see, A3 is the only one that has a "medium" score at the *purpose* commitment column. We evaluate its grade of commitment to Accessibility with a "medium" score because when analyzing its *purpose* statement, the approach is not focused on the Accessibility concern, but on a wide range of adaptation concerns --i.e. omnipresence, device independence, personalization, localization, privacy, etc.

Accordingly to the fact stated above at the *purpose* commitment column, we set A3 *assessment* column as "not specified", because the intent of this approach does not make any reference to a particularly Accessibility conformance criteria. On the other hand and since Accessibility is the main intent of A1, A2, A4, A5 and Ours, we set all the approaches' *purpose* commitment with a "high" score. A2 applies the W3C WCAG 1.0 for Accessibility conformance, and for that reason we set the approach's *assessment* column with the "WCAG 1.0" option. We set A1 *assessment* column with "other" because this approach applies its own practice to assess Accessibility instead of using a World-Wide reference guideline. A1 uses the DANTE tool [52] to extract visual objects from the page that support navigation. DANTE annotates the objects based on the Web Authoring for Accessibility (WAfA)[53] travel ontology. We set A4 *assessment* column with "generic" because this approach focuses on accessible design by using scenarios and guidelines, where "guidelines" means Accessibility standards or guidelines that contain interoperability techniques and heuristics for accessible design [52]. Finally, we set A5 and Ours *assessment* column with "WCAG 1.0 and 2.0". Both approaches originally were conceived to work with WCAG 1.0 checkpoints, but in [29], A5 shows how the proposal can work with WCAG 2.0. Also, we have already finished the migration of Ours to work with the W3C WCAG 2.0 success criteria.

At the *treatment* completeness column, A2 and A3 are the only ones that have "partial" scores but for different reasons. A2 aims to ensure an accessible Web page (or site) during a Web composition process that is managed by an authoring tool. We set a "partial" score at the *treatment* completeness column because the main focus of A2 is not placed on design issues but on evaluation to guarantee that no kind of new

[53] Web Authoring for Accessibility (WAfA) at http://augmented.man.ac.uk/ontologies/wafa.owl

Accessibility barriers can be introduced during a Web composition process. On the other hand, A3 completely illustrates how adaptation concerns can be added to an existing Hera-based Web application at the design level using aspect-oriented techniques. Despite to this fact, we also set a "partial" score for A3 at the *treatment* completeness column because the approach is not focused on adding Accessibility concerns. For A1, A4, A5 and Ours, the *treatment* completeness column is set with "full" scores and this is because these methods allow in different ways, early integration of the Accessibility in the design process. For example, A1 takes the WSDM design models as inputs --i.e. conceptual, navigation and implementation, and generates a set of annotations to improve Accessibility for visually impaired users. A4 defines a new way to take advantage of use cases, scenarios, test cases, personas, guidelines and checkpoints for Accessibility purposes during a design project employing a use case driven methodology. A5 follows the standard WCAG to model concepts and their relationships for AWA-Metamodel at the Compute Independent Model (CIM) of the MDA framework. Finally, Ours focuses on Accessibility requirements early taking advantages of applying AOSD principles to handle them properly as concerns during a Web development process.

Design criteria. Table 6.2 shows the resultant evaluation of the Design criteria. As we can see, we set the *paradigm* column for A1, A3, A5 and Ours as "main within MDSD" because these approaches show commitment and are fully identified with a particular *paradigm* to deal with Accessibility at design within different MDWE approaches. For example, at A1 the DANTE [52] annotation process uses a rule-based mapping model as a foundation *paradigm* to drive the authoring and mobility Accessibility annotations within WSDM [13]. A5 applies the MDA *paradigm* to define a domain-specific metamodel for Accessibility within the OOWS Navigational Model [18]. A3 and Ours apply consistently the AOSD *paradigm* when focusing on solving adaptation and Accessibility concerns, respectively. A3 adds aspect-oriented adaptation engineering to elements of the HERA Application Model [23], while Ours exploits the modeling capabilities of OOHDM Interface Models [36] to inject aspect-oriented Accessibility concerns identified at requirements elicitation. In the cases of A2 and A4, we set their *paradigm* column as "other" because they implement more than one

paradigm to deal with Accessibility. A2 applies a rule-based model as a foundation *paradigm* to drive the conditions under an accessible composition process takes place. But also, A2 proposes the Service-Oriented *paradigm* when using the Web Composition Service Linking System (WSLS) [20] as the authoring tool which enables the process of generating new and accessible Web content. Finally, A4 defines itself like tailored for design project employing a use-case driven methodology, so we say that A4 follows the Objet-Oriented *paradigm* but combined with a user profile-based technique called "Personas" [53].

Table 6.2: Design Criteria applied to the six approaches

DESIGN CRITERIA					
Approach	**Paradigm**	**Model**		**Technique**	
		Description	Completeness	Description / Name	Support
A1 **Plessers et al.** [35]	Main Within MDSD	Indentifies the interface elements, which may represent Accessibility barriers for visually impaired users, and annotates these interface elements with the semantic annotations.	Full	Yes Mapping rules established from the relationship between the concepts in the WSDM ontology and DANTE's WAfA ontology.	High
A2 **Centeno et al.** [9]	Other	Works on compositions, which are made of accessible chunks of HTML code, and evaluates these compositions with the compliance rules.	Partial	Yes Compliance rules established for Web compositions and formalized with W3C standards (XPath and XQuery expressions).	Medium
A3 **Casteleyn et al.** [6][7][8]	Main within MDSD	Selects the elements (units, attributes, relationships, etc.) from an HERA Application Model and injects these elements with the required Adaptation concerns.	Partial	Yes A domain specific language, baptized SEAL, which is custom-made to provide Adaptation support (through a set of constructs for aspects specification) in the context of Hera-S.	Medium
A4 **Zimmermann & Vanderheiden** [53]	Other	Models primary and secondary Personas to drive the user interface design for each use case.	Partial	No	
A5 **Moreno et al.** [29][30]	Main within MDSD	Defines several constructs in UML metamodel (MOF) to support the abstraction of Web Accessibility concepts based on WCAG standards.	Full	No	
Ours **Martin et al.**	Main within MDSD	Identifies Accessibility concerns in Web application requirements and maps them to widgets from abstract and concrete interface models using aspect orientation to meet the WCAG standards.	Full	Yes Three conceptual tools: ▪ UID with Integration Points, ▪ Association Tables, and ▪ SIG template for Accessibility that working together manage Accessibility concerns in an aspect-oriented manner.	High

Albeit for different reasons, A2, A3 and A4 have "partial" scores at the *model* completeness column. A2 is focused on formalizing the Accessibility conditions to be met by a Web composition of prewritten accessible chunks of Web pages, usually called "snippets". The approach proposes a set of Accessibility extra conditions for a range of possible Web compositions given a pair of accessible HTML snippets. We set a "partial" score for A2 at the *model* completeness column because the approach works over coarse-grained interface elements (existing accessible chunks composed of fine-grained elements as the raw material of the Web composition process) and, as a consequence, A2 focus its design effort on the evaluation over these coarse-grained elements. Also, it is a fact that the Service-Oriented paradigm is not inherent of the basic *model* (which is rule-based) but of the WSLS [20] proposed by the approach as the Accessibility enabled authoring tool for the *model's* implementation. A3 proposes a general *model* to extend an application with new functionality, considered as adaptation concerns, without having to redesign the entire application. We set a "partial" score for A3 at the *model* completeness column because the approach is focused on showing how the transformations required by an adaption concern can be specified independently from the original presentation level of a Web application using a generic transcoding tool. Therefore the *model* is not concerned on a detailed representation of the interface elements for an accessible design, but on showing how high-level support for adaptation specifications can be realized applying aspect-oriented techniques. A4 proposes a method that draws from the work on Accessibility guidelines and combines them with existing Object-Oriented techniques in software development. The approach encourages the early capture of Accessibility requirements using use cases, personas, scenarios and guidelines, and promotes manual/automatic testing based on test cases and Accessibility checkpoints (derived from guidelines) and expert reviews. In this case we set a "partial" score for A4 at the *model* completeness column because the proposed *model* does not represent these requirements into accessible interface elements at later stages of design. On the other hand, we set "full" scores for A1, A4 and Ours at the *model* completeness column. We set a "full" score for A1 at the *model* completeness column because the approach uses the DANTE's WAfA ontology to manage Accessibility of elementary interface elements for visually impaired users. The proposed *model* for the transformation process consists of two steps based on "authoring" and "mobility"

concepts and takes also into account the context of the journey --i.e. the purpose of the user's navigation. The conceptual knowledge captured at the WSDM design process is exploited by the *model* during the transformation because it provides mapping rules between modeling concepts in the WSDM ontology and the authoring concepts form WAfA ontology. A4 defines several meta-objects in MOF[54] to support the abstraction of Web Accessibility concepts and their relationships based on WCAG standards. Although A4 focuses its efforts on the meta-model, we set a "full" score for A4 *model* completeness column because the concepts provided by the approach can become concrete interface elements at the Platform Specific Model (PSM) for the MDA development process. Finally, we set a "full" score for Ours at the *model* completeness column because from the very beginning of the development process the approach focuses on identifying Accessibility requirements and managing them as AOSD concerns, consistently through abstract and concrete widgets of the OOHDM interface models. As a result of this proposal, the approach adds aspect-oriented Accessibility concerns early since requirement elicitation are weaved together using specialized techniques (for a thorough discussion on AOSD principles see [2][28]).

At the *techniques* support column, A4 and A5 do not propose any proprietary *technique* to complement themselves, since they apply existing design tools of software engineering and concepts from the MDA framework, respectively. As we can see at Table 6.2, A2 and A3 have "partial" scores at the *technique* support column. A2 offers a rule-based technique for a safe compound process delivering an accessible Web page from WCAG point of view. A2 has a "medium" score at the *technique* support column because the proposed technique is close to implementation and not focused on giving support to Accessibility design issues. Although the fact that A3 provides a domain specific language called SEAL[55], we set a "medium" score for A3 at the *technique* support column because the purpose of this proprietary custom-made language is to enrich the design level for adaptation support and not to reinforce the Accessibility

[54] OMG-MOF The Model-Object Facility at http://www.omg.org/mof/

[55] SEmantics-based Aspect-oriented Adaptation Language (SEAL) at http://wise.vub.ac.be/downloads/research/seal/SEALBNF.pdf

treatment. A1 and Ours have "high" scores at the *technique* support column. A1 provides mapping rules between the concepts in the WSDM ontology and DANTE's WAfA ontology which enable enriching the design level to reinforce the Accessibility propose by taking the WSDM conceptual models as input and annotating them with authoring and mobility concepts. Finally, Ours provides the User-Interaction Diagram (UID) with Integration Points and the Softgoal Interdependency Graph (SIG) template for Accessibility linked by the Association Tables. We set a "high" score for Ours at the *technique* support column because these conceptual tools where specially developed to provide aspect-oriented support at the design level for Accessibility purpose.

Table 6.3: Other Criteria applied to the six approaches

OTHER CRITERIA			
Approach	**Background approaches**		**Supporting tool**
	Name	Purpose	
A1 **Plessers et al.** [35]	DANTE [52]	Used to perform the semantic annotation process of Web pages.	Yes Implements WSDM-DANTE mapping rules to automatically generate semantic annotations.
A2 **Centeno et al.** [9]	WSLS: A Service-based System for Reuse-Oriented Web engineering [20]	Used as the Accessibility enabled authoring tool.	Yes Shows for some selected rules (based on automatable WCAG checkpoints) how WSLS can afford compliance to these rules.
A3 **Casteleyn et al.** [6][7][8]	Component-based AMACONT framework [15][16] [32]	Used as the first implementation of a presentation engine for HERA-S.	Yes Integrates SEAL in HydraGen system, which is the latest implementation generation tool for Hera-S.
A4 **Zimmermann & Vanderheiden** [53]	Use Cases and Personas	Applied to model user profiles linked to their Accessibility requirements.	No
A5 **Moreno et al.** [29][30]	MDA framework	Applied to support AWA for MDA development process.	Yes Provides AWA-MetamodelEditor for graphical support to AWA-Metamodel.
Ours **Martin et al.**	User Interaction Diagram (UID) for modeling user-system interaction [43] Softgoal Interdependency Graph (SIG) for modeling non-functional requirements (NFRs) [11][12]	Extended for supporting Accessibility requirements.	Yes Provides a supporting tool to discover crosscutting concerns and apply aspects at the Abstract User Interface model.

Other criteria. Table 6.3 shows the resultant evaluation of the Other criteria. At the *background approach* column, we can see that all the proposals have previous works

and these works are fundamental pieces to the operation of the approaches. A1 founds its work on DANTE's WAfA ontology [52] that is applied to enhance the mobility of visually impaired Web users by providing screen readers with extra knowledge to better facilitate the audio presentation of the Web page. A2 uses the WSLS system [20], which is a component-based system applying the service-oriented paradigm to compound, discover and reuse services. The GAC transcoder [16] provided by the ANACONT framework [15] is foundational to A3, since this approach exploits a transcoding tool for making Web application adaptive. A4 applies uses cases and scenarios extended with the "personas" profiling technique for describing Accessibility interfaces' needs and other usage requirements of users with disabilities.

As we can see in Table 6.3, A4 is the only one that has "No" at the *supporting tool* column, while A1, A2, A3, A5 and Ours offer at least some kind of executable implementation of their proposals. A1 presented a combined approach where the mapping rules between the WSDM [13] concepts and the DANTE [52] concepts are implemented. This implementation allows about +/- 70% of the DANTE concepts annotations to be generated automatically without any extra effort from designers. A2 extends the WSLS system [20] to afford compliance to a set of selected rules that guarantee accessible Web composition. The tool seems to give already some promising results since the fact that the WSLS framework is implemented on the top of the .NET framework and gives support to XML technologies. A3 offers a latest implementation of the approach generation tool for HERA-S that integrates SEAL in HydraGen engine[56] (an implementation generation tool for Hera-S developed externally by the University of Eindhoven), to show their adaptation engineering perspective applying pointcuts and advices expressions. A5 provides the AWA-MetamodelEditor for graphical metamodel support that is based on the Graphical Modeling Framework (GMF)[57]. Finally, Ours provides a tool at Stage 3 of the proposed development process that helps designer and developers to produce accessible interfaces by moving from abstract to concrete architectural views using aspect-orientation --i.e. discovering

[56] Hydragen: An implementation of Hera-S at http://wwwis.win.tue.nl/~ksluijs/material/Singh-Master-Thesis-2007.pdf

[57] The Eclipse Graphical Modeling Project (GMP) at http://www.eclipse.org/modeling/gmp/

crosscutting concerns and applying aspects at the abstract user interface model from knowledge about Accessibility obtained in previous stages. Related to Ours, it is also important to highlight that as we have already indicated in Chapter 4 and later, we have showed with the case study in Chapter 5, there are cases in which we can develop artifacts once and then reused them, as we required. The reuse capabilities of Ours is a main advantage, because propitiates the supporting tool to have a design artifacts repository. For example, and as we have showed in Figures 5.3, 5.4 and 5.5, the Accessibility softgoal for the HTML *image* element can be modeled once and then applied for the SIG instantiation any time is required.

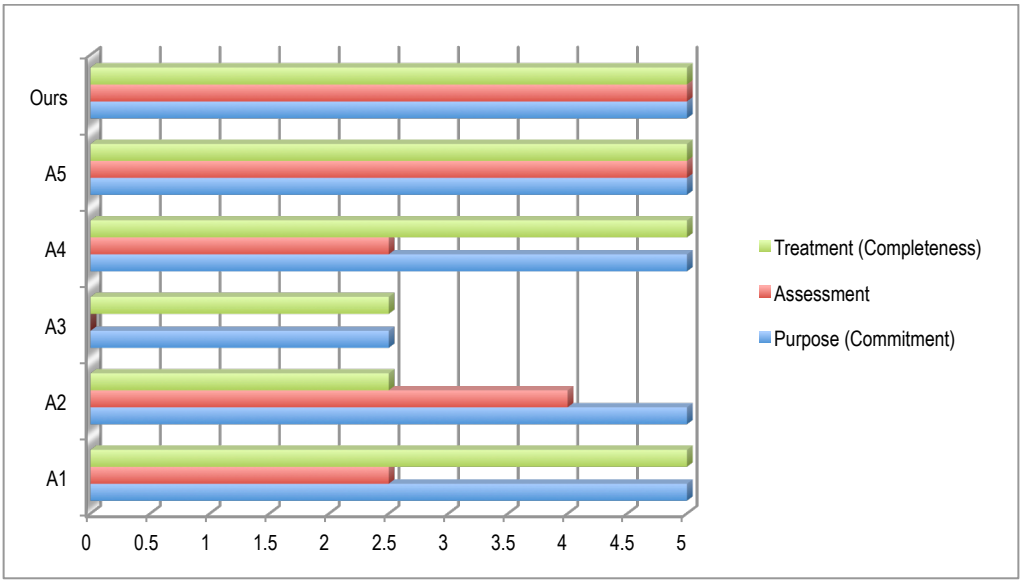

Figure 6.2: Scoring the six approaches for the Accessibility Criteria

To summarize the results of the six approaches' comparison, we score the topics related to the Accessibility and Design criteria from 0 to 5, as follows: (i) the scores "high" and "full" match to 5, while the scores "medium" and "partial" match to 2.5; (ii) at the assessment topic, the option "WCAG 1.0 and 2.0" matches to 5, the option "WCAG 1.0" matches to 4, the option "generic" and "other" match to 2.5, and the option "not specified" matches to 0; and finally (iii) at the paradigm topic, the option "main within MDSD" matches to 5, while the option "other" matches to 2.5. Figures 6.2 and 6.3 show the scoring of the six approaches for the Accessibility and Design criteria, respectively.

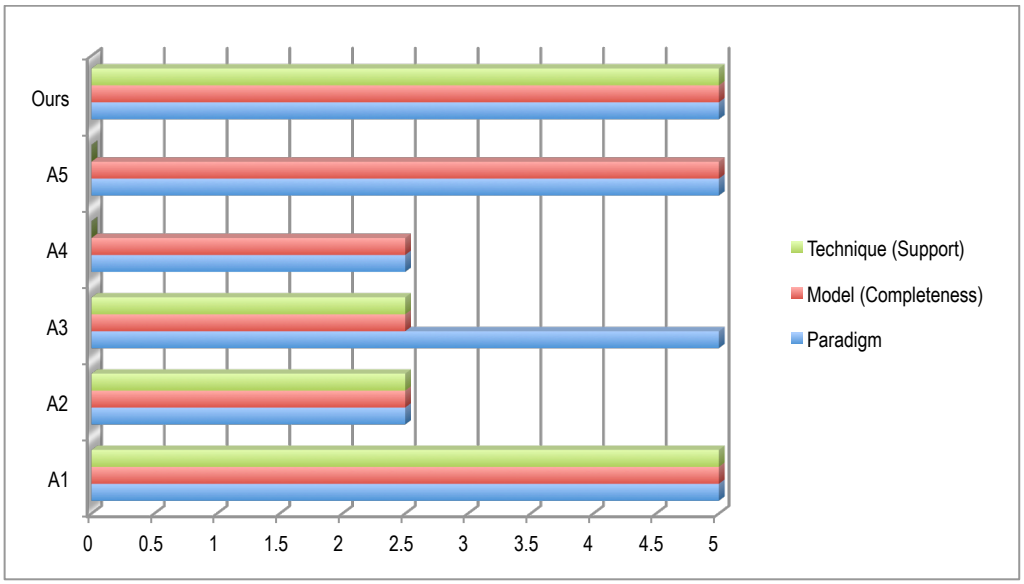

Figure 6.3: Scoring the six approaches for the Design Criteria

To complete this summary, Figure 6.4 shows the average of scores for the six approaches by Criteria. We should note that for the Other Criteria, we score only the *supporting tool* topic by simply matching the options "yes" and "no" to 5 and 0, respectively.

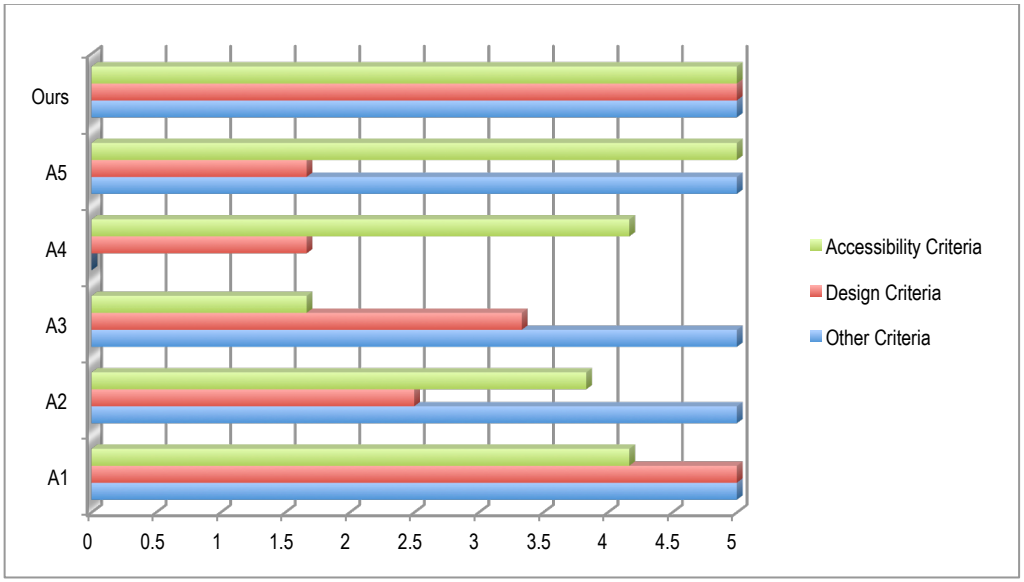

Figure 6.4: The average of scores for the six approaches by Criteria

6.3 Focusing on Ours

We dedicate this Section to provide some extra discussion about our proposal. As we already said, Ours allows developers to produce accessible interfaces by moving from abstract to concrete architectural views using aspect-orientation. This is a main advantage, since allows developers to keep in mind a clear picture of how these architectural views relate each other during the development process, while preserving their own properties: (i) the abstract view ensures clean designs --i.e. free of crosscutting symptoms, which are separated and modeled as aspects for their modularization; while (ii) the concrete view provides the implementation of these designs, but as a consequence of the weaving process that takes place at the code level. Thus, Ours uses aspect-orientation to propose a smooth and open transition between models (abstract and concrete views), since this transition allows the independence of the way clean designs will be implemented into accessible code.

At this point, we revisit the argument, which we stated when applying Ours in Section 5.2, to the case study in Section 5.1, about alternatives in the navigation path. As Figure 5.1 (d) shows, the case study offers the student two pages to help to the login process in Figure 5.1 (c). We highlighted that browsing these pages is optional and therefore, if the student follows these help links, his/her decision will produce a different navigation path. As we said before, we focus on the UI models because, undoubtedly, is at the UI where Accessibility barrier finally show, but notice that this is one of those cases in which navigational issues can affect Accessibility. This is the reason why, to improve the user's experience when browsing to achieve the desired functionality, we have to consider the UI designs for each alternative in the navigation path we have defined as important for the task's functionality. This means that if we provide the user with alternatives in the navigation path, they must be explored and modeled before properly, because they can be relevant to Accessibility and therefore to the success of the user's task. This is an advantage of Ours, because although Ours is focused on UI models, also allows to explore navigational models to avoid unexplored optional browsing that can lead to user interfaces which were not considered initially.

As Figure 6.5 shows, this is possible mainly because of two reasons. In first place, the UID is the conceptual tool used by OOHDM to state transformations between Web application requirements (use case model) and the conceptual, navigational and interface models. As Figure 6.5 shows, this is the same principle that Ours propitiates between Web applications requirements and accessible UI models. Ours uses two conceptual tools (the UID with *integration points* and SIG *template* for Accessibility), with which the interaction between OOHDM models links and reinforces Accessibility needs.

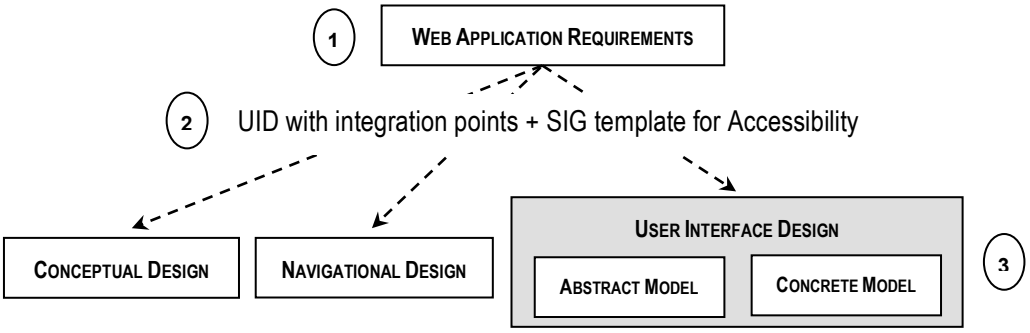

Figure 6.5: Ours within MDSD paradigm

In second place, since Ours is conceived within the MDSD paradigm, models are related to each other and as a consequence of an iterative and incremental development process. Thus, Ours allows: (i) going back from UI models to navigational models to look for alternatives in the navigation path, (ii) assessing the need and relevance of these alternatives to the functionality under develop, and (iii) going forward from navigational models to UI models to check the Accessibility of the UI related to these alternatives.

6.3.1 Migrating to WCAG 2.0

We have already given part of our motivation for applying WCAG 1.0 [45] instead of WCAG 2.0 [46] in Section 3.6.

In first place, and to avoid linking the selection of the WCAG 1.0 only to issues related to the adoption rate in the world, it seems appropriate to highlight that as we are concerned with Accessibility, we have a few quibbles about the decision made on the usefulness of certain checkpoints in the WCAG 2.0 document.

Table 6.1: Association Table for the HTML Control Elements Group using WCAG 2.0

ASPECT	ONTOLOGY WIDGETS (ABSTRACT WIDGETS)	HTML ELEMENTS (CONCRETE WIDGETS)		WCAG 2.0 SUCCESS CRITERIA AND THEIR LEVELS OF CONFORMANCE: [A] [AA] OR [AAA]			DESIGN DECISION CLASS related to USER-APPLICATION INTERACTION
				2.4.3 [A]	1.3.1 [A]	4.1.2 [A]	
				D-P 🛠	P	D-P	DIALOG (D) PRESENTATION (P) PRAGMATIC 🛠
I. TSCONTROL SIG'S USER TECHNOLOGY SUPPORT BRANCH	INDEFINITEVARIABLE	TEXT FIELD	INPUT TEXT…	✓			
		TEXT AREA	TEXTAREA…	✓			
		RELATED CONTROLS	FIELDSET…				
	PREDEFINEDVARIABLE MULTIPLECHOICES	CHECK BOX	INPUT CHECKBOX…	✓			
		MULTIPLE OPTION MENU	SELECT MULTIPLE…	✓			
		RELATED OPTIONS	OPTGROUP…				
	PREDEFINEDVARIABLE SINGLECHOICES	RADIO BUTTON	INPUT RADIO…	✓			
		SIMPLE OPTION MENU	SELECT…	✓			
II. LSCONTROL SIG'S USER LAYOUT SUPPORT BRANCH	INDEFINITEVARIABLE	TEXTFIELD	INPUT TEXT…		✓	✓	
		TEXTAREA	TEXTAREA…		✓	✓	
		RELATED CONTROLS	FIELDSET…		✓	✓	
	PREDEFINEDVARIABLE MULTIPLECHOICES	CHECKBOX	INPUT CHECKBOX…		✓	✓	
		MULTIPLE OPTION MENU	SELECT MULTIPLE …		✓	✓	
		RELATED OPTIONS	OPTGROUP…		✓	✓	
	PREDEFINEDVARIABLE SINGLECHOICES	RADIO BUTTON	INPUT RADIO…		✓	✓	
		SIMPLE OPTION MENU	SELECT…		✓	✓	

For example, WCAG 1.0 provides the checkpoint 12.3 which basically states that the information should be grouped to divide large blocks of information into more manageable groups and this is especially true for the HTML *related controls* element (a set of HTML *text field* elements). The WCAG 2.0 version from January 2006 was also clear on this point, providing the criterion 4.1.3, which basically says that the label of each user interface control in the Web content that accepts input from the user can be

programmatically determined and explicitly associated with the control. Unfortunately, success criterion 4.1.3 has been removed and WCAG 2.0 relies on success criterion 1.3.1 to cover the labeling of related controls, which is not explicit enough to ensure the absence of this important accessibility barrier. In this sense, we fully agree with the statement about the WCAG 2.0 on [41]: “not having any success criteria specifically dealing with forms is certainly a mistake”.

However, aware that the new guidelines and the move to technological neutrality are undoubtedly good, we don’t see major inconveniences to upgrade our approach to WCAG 2.0 when necessary. As we discussed before, our approach is based on the use of UIDs with integration points and the SIG template for Accessibility linked by association tables. These conceptual tools are able to support the success criteria from WCAG 2.0 instead of checkpoints from WCAG 1.0 applying some straightforward redefinitions and adjustments. As an example, Table 7.1 shows the association table for HTML control elements group using WCAG 2.0 success criteria. We highlight that to realize this upgrade we use the comparison provided by W3C-WAI in [49], since there are still some discrepancies at the Accessibility community[58] when providing mappings between the WCAG 1.0 checkpoints onto the WCAG 2.0 success criteria.

[58] Examples of these comparisons at http://www.w3.org/WAI/WCAG20/from10/comparison/; http://wipa.org.au/papers/wcag- migration.htm; http://www.usability.com.au/resources/wcag2./