通过模型驱动工程满足老年用户与年龄相关的无障碍需求

Shavindra Wickramathilaka软件系统与网络安全 莫纳什大学澳大利亚墨尔本
Shavindra.
Wickramathilaka@monash.edu

Ingo Mueller软件系统与网络安全 莫纳什大学墨尔本，澳大利亚
Ingo.Mueller@monash.edu

arXiv:2305.00695v2 [cs.SE] 5 May 2023

摘要——**老年人在使用软件应用程序时面临无障碍障碍的主要原因之一，是软件用户界面未能妥善满足老年人因年龄增长而产生的用户界面（UI）需求（例如身体和认知限制）。现有文献提出，基于模型驱动工程的用户界面适配是解决这一现象的重要方案。然而，在对该领域进行探索的过程中，我们发现，与《网页内容无障碍指南》（WCAG）等广受认可的无障碍标准相比，现有研究在将无障碍性融入软件建模工具和方法方面缺乏全面性。因此，本文提出了一项研究路线图，旨在以 WCAG 作为参考框架，设计特定领域语言，通过用户情境信息和用户界面适配规则建模，对老年用户的多样化无障碍场景进行建模，从而满足 WCAG 所规定的无障碍标准。**

索引词—**老年人、软件无障碍、自适应用户界面、模型驱动工程**

I. 引言

根据世界卫生组织（WHO）的数据，到 2030 [1]，每六个人中就有一人年满六十岁。尽管这一趋势日益明显，开发人员却忽视了满足老年目标受众的软件无障碍需求，导致了数字排斥做法 [2]。诸如 [2]–[4] 的研究表明，导致老年人面临数字鸿沟的若干关键挑战可归因于用户界面（UI）。这些界面往往未能体现老年人所经历的与年龄相关的身体和认知挑战的多样性，例如视力和听力障碍、精神运动障碍以及记忆力限制。因此，无障碍障碍妨碍了老年人与其软件应用程序的交互 [5]。

因此，软件开发人员必须专门针对老年人进行设计，并满足他们的无障碍需求。然而，由于每位老年用户与年龄相关的需求本身具有多样性和可变性，这并非易事。换言之，在设计时并不总能全面了解所有无障碍需求。此外，根据 [6]，从一开始就手动将无障碍功能融入用户界面可能十分复杂，且成本效益不高。

一种有助于解决这一问题的重要概念是模型驱动工程（MDE） [7]。模型驱动工程（MDE）将软件模型视为生成应用程序代码的主要制品，整个过程几乎不需要手动编写代码 [7]。正如我们在第三节中介绍的多项研究 [5], [8]–[10]所述，领域特定语言（DSL）软件模型使用户界面设计师乃至领域专家能够全面、正式地表达与年龄相关的用户界面需求。这些领域特定语言模型可以在设计时根据最终用户的需求进行定制，并输入模型驱动工程（MDE）的代码生成流程，以生成适应性强且易于访问的用户界面。

除了模型驱动工程的自动代码生成能力之外，当我们将用户界面适配能力作为模型驱动工程流程的一部分进行建模时，它尤其有助于我们解决这一问题。由于与年龄相关的用户界面需求多变且各不相同，我们至多只能通过设计时的模型捕捉老年人需求的一个快照，例如以个人、设备和环境相关的上下文信息形式呈现。然而，上下文维度随时可能发生变化；一旦发生变化，我们面对的就是模型中未涵盖的上下文场景。应对这类上下文变化的一种方式是在运行时和设计时将用户界面适配技术集成到模型驱动工程流程中，从而相应地调整模型并生成适配后的用户界面实例。根据一项 2014年文献调查 [6], , 这是应对用户界面上下文可变性的主要方法。

本文探讨了通过可适配的模型驱动工程方法满足老年人无障碍需求的最新进展。我们首先在第三节报告了一项文献调查，表明近年来这一主题已受到关注。然而，与普遍认可的无障碍开发标准（如万维网联盟（W3C）的 WCAG [11]）相比，大多数方法仍不完善。尽管一些方法将 WCAG 及其他相关标准纳入模型驱动工程工具，但我们对选定论文的深入考察表明，现有建模方法和适配技术的能力仍有多处改进空间（见第四节）。其次，我们在第五节提出研究路线图，旨在创建并评估一种模型驱动的无障碍自适应用户界面建模与代码生成工具，以弥补特定无障碍需求方面的不足。

II. 背景

A. 无障碍性

在信息技术领域，“无障碍性”是指“硬件和软件对任何潜在用户的操作适用性。” [12]。

- *Web* 内容无障碍指南（*WCAG*）：WCAG 提供了一套广为人知的建议，旨在提升网页内容对残障人士的无障碍性，包括听力障碍、视力障碍和认知障碍人士。该标准由原则、指南、成功标准和建议性技术构成一个层级体系 [11]。

B. 用户界面适配

用户界面适配可定义为系统或用户调整用户界面的能力 [5]。用户界面适配的示例包括布局、导航、样式和输入/输出模态变化。

- 设计时用户界面适配：指用户界面开发过程中进行的用户界面设计。开发人员根据已知且静态的使用情境信息设计和定制用户界面模型，随后可使用这些模型生成最终用户界面。
- 运行时用户界面适应：发生在应用程序的最终用户界面运行期间。这些适应可因使用情境场景动态变化或用户可调整的配置而发生。
- 使用情境（*CoU*）：Calvary等人 [13]将使用情境定义为一个包含以下实体的n元组：1）系统的**用户** 及其属性（例如功能障碍和语言偏好）；2）**平台**：用户用来与系统交互的硬件和软件组件（例如设备类型和屏幕尺寸）；3）**环境**：用户与系统发生交互的环境（例如环境噪声和天气）。在适配用户界面时，可使用使用情境来建模以用户为中心的无障碍场景。

C. 模型驱动工程（*MDE*）

MDE是一种软件开发方法论，在这种方法中，软件模型被视为软件开发过程的主要制品。将模型传入模型驱动工程流程后，它们会自动转换为代码制品。 [7]

- 领域特定语言（*DSL*）：领域特定语言是一种软件设计和/或开发语言，能够表达应用领域特定概念。领域特定语言的表达能力和表示法使分析人员和领域专家能够向开发人员传达领域需求。 [7]

- 抽象用户界面模型（*AUI*）：一种用于呈现领域特定语言中所阐释的用户界面领域概念的模型，不受目标平台或交互方式的限制。抽象用户界面模型的类似术语是“平台无关用户界面模型”。 [13]

- 具体用户界面模型（*CUI*）：CUI 根据平台和交互模式，将用户界面领域概念呈现出来。在模型驱动工程中，CUI 模型是将抽象用户界面转换为平台特定模型的结果，它明确界定了最终用户界面的外观和体验。 [13]
- 最终用户界面（*FUI*）：在模型驱动工程中，最终用户界面是通过转换 CUI 模型生成的源代码。随后可将其解释或编译为可执行用户界面。（某些方法需要开发人员手动补充代码，才能使生成的代码可执行。） [13]

III. 相关工作

本节简要回顾自适应用户界面和基于模型驱动工程的自适应用户界面领域的重要文献。文献综述的筛选标准为：1）采用基于模型驱动工程的方法；2）涉及用户界面适配；3）近期发表的研究，即 2010年之后发表的研究。

A. 仅支持设计时用户界面自适应的模型驱动工程方法

在一项近期（2022年）的研究中，Braham等人 [14] 提出了一种模型驱动工程方法，可在设计时生成自适应用户界面，以应对普适计算应用中的异构使用情境。为此，他们使用不同的本体对用户界面、使用情境和适配规则进行建模。后两种模型会根据无障碍性场景对用户界面模型进行自适应。论文指出了若干局限：1）上下文本体元模型未涵盖设备相关上下文信息，例如平台、设备、网络和屏幕尺寸。现有信息不足以确定本体模型对用户和环境上下文建模的能力。2）给出的自适应规则元模型不足以对有条件的复合适应规则和模态变化（例如屏幕阅读器、文本转语音）进行建模。此外，用户界面布局自适应还可扩展至包含边框、外边距、内边距和字体等样式更改。

Bendaly Hlaoui等人，[5]提出了一种模型驱动工程方法，该方法仅支持设计时用户界面适配。不过，与之前的研究相比，他们的使用情境和用户界面适配建模方法相对更加成熟，而且他们的工作同样受到残障用户的用户界面适配需求所驱动。使用情境场景基于一种新颖的本体进行建模，该本体提供残障、用户、设备和环境背景信息。随后，将本体实例模型中的元素标注到软件未自适应的抽象用户界面模型（AUI）中，以生成适配后的AUI模型。通过对盲人、低视力用户和失聪用户开展用户研究评估，作者发现，计算平台使用经验较少的参与者对其适配后的用户界面满意度较低。虽然作者没有进一步详细讨论这一发现，但他们确实建议根据用户的计算机使用专长提供用户配置选项。因此，扩展该方法的一个关键机会是向最终用户提供运行时用户界面配置选项。向用户提供的配置选项级别取决于其计算机使用经验水平。

在另一项研究中，Bacha 等人 [15] 提出了一种模型驱动工程方法，通过在设计时进行适配来实现用户界面个性化。

这通过三种建模制品实现：1）用于捕获使用场景上下文的上下文模型；2）用于捕获并建模目标应用程序领域信息的领域本体模型；3）用于描述任务及其间信息流的业务流程模型。这三个模型之间的映射使设计人员能够生成抽象用户界面（AUI）模型，随后在模型驱动工程流程中将其转换为最终用户界面。他们研究的一项局限是，尽管作者强调用户界面个性化，但并未考虑是否符合用户界面可访问性标准（例如 WCAG）。

不同于前述研究，Park and Lee [16] 提出了一种需求工程（RE）方法，在记录自适应用户界面需求时引入了模型驱动工程（MDE）的三个抽象层次。该方法包含三层需求工程表示法：1）抽象用户界面需求：用于表达高层次的、领域/平台无关的自适应用户界面目标/需求。2）具体用户界面需求：进一步细化抽象用户界面需求，纳入领域/平台特定的功能性和质量性用户界面需求。3）最终用户界面需求：进一步详细说明需求，纳入触发规则、传感器类型和可用性指标等实现细节。由于缺乏针对需求工程从业者的用户研究，读者难以理解该概念性方法的有效性。该方法的另一个局限在于，未讨论超出需求工程阶段的内容。所提出的自适应用户界面需求捕获表示法必须与模型驱动过程集成，以将已定义的需求转换为可执行用户界面。

B. 支持运行时用户界面适配的模型驱动工程方法

Yigitbas 等的 [10] 2020 研究提出了两个新颖的领域特定语言（ContextML 和 AdaptML），并将其与交互式流程建模语言（IFML）集成。IFML 是一种 OMG 标准，支持对用户界面 [17], 进行平台无关定义；该研究使软件能够在运行时自动适配用户界面。ContextML 对使用情境场景进行建模，而 AdaptML 则对适配规则进行建模，这些规则同时引用上下文模型中的使用情境元素和以 IFML 指定的抽象用户界面模型。由于其元模型的限制，ContextML 在对用户上下文元素进行建模方面存在局限。例如，ContextML 无法根据用户的熟练程度或对某种语言的熟悉程度，或其语音输出模态所使用的口音/方言，来定义使用情境场景。同样，AdaptML 的适配能力也受到限制。例如，文本替代项、基于时间的媒体以及定时活动（即超时、中断、重新身份验证）等若干 WCAG指南，均未涵盖在 AdaptML 元模型中。

Ghaibiet 等人 [8]提出了一种用于生成对使用情境敏感的自适应用户界面的模型驱动工程方法。在设计时，开发人员可以使用图形化模型驱动工程工具包来管理使用上下文参数、管理适配规则并生成定制用户界面。在运行时，用户还可以自由地为每个使用情境场景创建自己的用户界面适配规则，对其进行管理，并生成适应后的用户界面实例。该论文介绍了若干局限性：1）使用情境元模型较为浅显，无法支持深入且全面的使用情境场景建模。例如，原型示例表明，建模时仅涵盖了有限的情境参数，而语言、年龄、确切的屏幕尺寸、环境光/噪声以及语音转文本模态等参数均未被考虑。2）元模型不支持用户界面适应逻辑，例如复合适应规则和布局相关用户界面适应（即字体类型、外边距和边框）。这限制了遵循 WCAG 中所提出的用户界面可访问性指南的可能性。

Akiki 等人 [9] 提出了一种基于角色的用户界面适应机制，用于应对各种使用情境（CoU）场景。该方法的模型驱动工程元模型仅支持图形化/布局用户界面适应，不涵盖文本转语音、语音识别和仅使用键盘进行导航等多模态（听觉和触觉）功能。用户界面适配本身通过采用基于角色的访问控制（RBAC） [18]中的元素来实现。在该方法中，用户界面元素和用户界面适配规则被视为可访问性资源，并以“角色”的形式分配给用户账户。在设计时，用户账户会被分配与残障和文化等使用情境信息相关联的角色。但由于某些情境信息（例如环境亮度和噪声级别）只有在运行时才能获知，因此，当用户界面适应工作流触发后，与后一类情境信息相关联的角色会被动态分配给用户账户。通过加入多模态功能，可以扩展该方法，从而提升适配后用户界面面向视力障碍者和老年人等用户群体的无障碍性。另一种改进方向是为模型驱动工程用户界面元模型添加导航适应支持，以进一步确保符合 WCAG合规性。

Akiki [19], 进一步拓展了这项工作，并提出了自适应界面的情境帮助（CHAIN）。这为模型驱动的自适应用户界面系统提供了一种互补方法，用于开发涵盖不同用户界面适配场景的情境感知用户界面帮助。用户界面帮助模型会被转换为 CHAINXML 文档，并与模型驱动工程用户界面模型分开保存。帮助用户界面元素与运行中的用户界面元素之间通过映射建立关联。这样一来，即使发生导航、布局、呈现方式以及基于小部件的变化（调整大小、重新定位和更改用户界面小部件），运行中的用户界面也能保留其帮助功能。这项研究的局限之一是，情境帮助用户界面叠加层无法充分感知软件用户界面如何进行自适应。例如，由于这一局限，当用户界面在运行时适配为纯语音输入/输出模态时，CHAIN 帮助用户界面无法切换为纯语音帮助。

Bongartz 等人 [20]提出了一种支持工作环境中的模型驱动的自适应用户界面的方法，重点关注多模态。需要注意的是，论文中对“工作环境”一词的定义较为宽泛，但该方法通过仓库拣选系统的具体示例进行了说明。在设计时，开发人员可以使用用户界面定义语言 MARIA [21] 指定具体用户界面模型的三个版本（语音、图形化和多模态）。开发人员

还可以通过基于 XML 的领域特定语言定义一组适配规则，以根据给定的使用情境（CoU）场景变化调整具体用户界面模型。在运行时，系统会监听使用情境的变化，并执行适配规则来定制用户界面。就该研究的局限性而言，开发人员需要为同一个用户界面模型设计三个版本，导致重复投入。模型驱动工程流程可以进一步完善，改为使用单个抽象模型，根据模态模式生成具体用户界面模型实例。另一个值得注意的问题是，该研究将使用情境管理模块解释为一个黑箱组件。因此，研究提供的信息不足以阐明自适应用户界面系统的情境建模能力。同样，研究也缺少有关适配规则领域特定语言元模型的信息，因此其用户界面适配能力较为模糊。

Minon 等人 [22] 提出了一种名为适配集成系统（AIS）的工具，该工具为符合 CAMELEON 参考框架[13] 所述用户界面抽象层级的模型驱动工程工具提供适配规则集成扩展。AIS 旨在通过设计时和运行时自动调整用户界面，以满足有特殊需求的用户群体（例如视力、听力或认知障碍者）的用户界面可访问性要求。AIS 还包含一套针对上述用户群体的用户界面适配规则。设计时，模型驱动工程用户界面工具的设计人员手动输入一个属于 CAMELEON 框架任一抽象层级的用户界面模型，以及用于指明用户残障情况的参数。随后，设计人员将收到根据用户残障情况适配后的用户界面模型。为说明模型驱动工程用户界面工具与 AIS 在具体用户界面（CUI）层级的运行时适配集成过程，Minon 等人举例如下：当模型驱动工程用户界面工具检测到情境变化时，会将情境事件和该工具的 CUI 模型发送至 AIS。之后，AIS 根据请求参数匹配并选择合适的适配规则，并将其应用于 CUI 模型。最后，模型驱动工程用户界面工具的最终用户界面生成器接收已适配的 CUI 模型，而非原始模型，并生成适配后的最终用户界面 [22]。AIS 的一个局限在于，其适配规则未能全面涵盖 WCAG 的无障碍需求（即导航和定时活动适配）。

IV. 研究空白

第三节中审阅的每项研究都可以通过用户界面适配解决老年人的部分无障碍问题。然而，我们对每种方法进行深入评估后发现，它们各自也存在可访问性限制，尤其是与 WCAG 等标准化指南集逐项对照时。例如， [10] 的方法是第三节中最新、最成熟的方法之一，但并未考虑一些无障碍功能，例如时间敏感活动适配（如调整定时活动、超时和重新身份验证的时长），也未支持用户的语言熟练度。同样， [22] 也存在一些问题，例如缺乏对导航适配的支持（如：修改导航路线以绕过区块，并提供多条路径）。表 I 根据现有研究提供的信息，总结了我们发现的主要无障碍差距（参照 WCAG）。有些研究提供的信息不足，例如缺少元模型、对方法论的深入说明，或示例／原型，因此无法全面识别其中的所有无障碍差距。

表 I 第三节中的主要无障碍功能缺口

自适应用户界面的无障碍功能缺口	**论文**
多模态支持（例如：文本转语音、语音转文本和屏幕放大镜）	[8]–[10], [14]
用户界面导航支持（例如：绕过区块、修改.....）原生路线和多条路线）	[9], [22]
呈现方式/样式/布局适配支持（例如：边框、外边距、内边距和字体)	[14]
平台相关支持 可变性（例如：屏幕尺寸和设备类型）	[14]
针对时间敏感型用户界面活动的适配（例如：定时活动、超时和重新身份验证）	All
错误处理和用户界面帮助	除 [19]之外的所有内容
对语言熟练度相关的用户界面适配的支持（例如：跨语言支持、阅读理解、习语和行话）解、习语和行话）	All

即使是通过用户界面适配对无障碍场景进行建模方面已经较为成熟的方法，例如 [5], [9], [10], ，在针对老年用户的自适应用户界面支持方面仍存在严重局限。我们认为，许多人以临时、零散的方式考虑目标终端用户的无障碍需求，而没有参考框架，从而导致某些领域覆盖不足。此外，即使是提到纳入无障碍标准的研究，例如 [5] 和 [22],，也仅在少数情况下这样做，例如与文本大小以及背景与前景对比度相关的指南。

因此，支持用户界面适配的应用程序的开发人员需要考虑“无障碍优先”范式，在该范式下，最终用户对自适应无障碍用户界面的需求在应用程序设计过程中被视为首要需求。只有这样，他们才能为类似老年人的用户群体提供完整的无障碍支持，因为这类用户与年龄相关的用户界面需求多样且不断变化。

为了实现这一目标，我们可以借鉴现有研究中一些最有前景的模型驱动工程用户界面适配方法。根据我们的文献综述，现有方法通过以下两种技术融入了无障碍性。首先，使用情境场景模型利用用户、平台和环境上下文维度，描述用户使用应用程序时可能面临的无障碍情境。其次，引用各个使用情境场景的用户界面适配规则用于执行具体的用户界面适配。

V. 研究路线图

为了制定无障碍优先方法，以开发模型驱动的自适应用户界面，我们认为，在后续及未来工作中应遵循以下目标和任务。

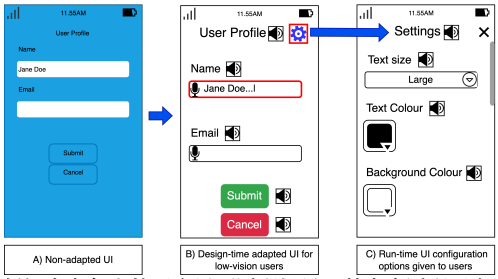


图1. 面向低视力老年人的用户界面适配场景，其中未适配用户界面（A）通过用户界面修改，在设计时（B）/运行时（C）进行适配，包括对比度、文本大小和多模态（文本转语音和语音转文本）的调整。

A. 我们的目标

一种基于模型驱动工程的自适应用户界面建模与代码生成工具，旨在帮助软件开发人员构建满足老年人年龄相关需求的无障碍软件用户界面。

B. 关键目标

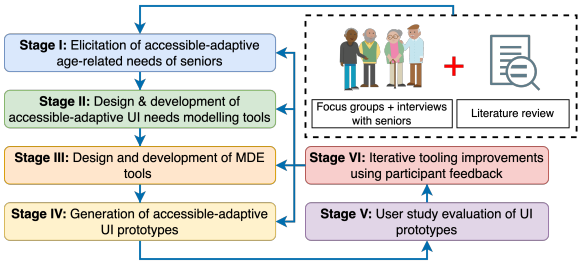


图 2。研究路线图

1) 我们如何从老年人那里收集一套具体的无障碍性和自适应用户界面需求?

需求获取通过两种信息来源进行：1) **焦点小组研究**以及对多组老年用户进行的后续**访谈**；2) 持续开展关于年龄相关用户界面无障碍需求的文献综述。我们正在开展这两项活动，以便为面向老年终端用户的自适应、基于模型驱动工程的用户界面生成丰富而完整的需求集。

2) 如何以符合 WCAG 的方式对老年人的无障碍需求进行建模?

我们认为，需要两种新的符合 WCAG 的**领域特定语言**来对上一步提取的需求进行建模：1）一种用于对使用情境场景进行建模的图形化领域特定语言；2）一种用于指定用户界面适配规则的文本领域特定语言。我们借鉴了[10]的 ContextML 和 AdaptML 领域特定语言，它们具有类似用途。“WCAG 合规性”意味着，每一种代表所提取需求中无障碍场景的用户—平台—环境使用情境组合，以及与之相关的适配规则，都能够满足 WCAG 成功准则。如果 WCAG 不足以涵盖所有已识别的无障碍需求，我们还会参考现有的无障碍领域文献，尤其是人机交互领域的文献。

3) 如何设计和开发模型驱动工程流程以生成最终用户界面?

我们的目标是将步骤 2 中的使用情境模型和适配规则模型纳入模型驱动工程流程，使这些模型能够将未适配用户界面模型转换为已适配用户界面模型。该模型驱动工程流程应具备以下两项能力：1）**设计时用户界面适配**——开发人员可以利用最终用户的使用情境信息，在设计时调整用户界面，使其符合该用户所属的用户画像；2）**最终用户可在运行时个性化用户界面** ——我们认识到，开发人员与老年人差异很大，因此，要让开发人员在设计时捕捉老年人多样化的与年龄相关的用户界面需求颇具挑战性。因此，我们计划为老年用户提供相应的方式，让他们能够根据个人偏好自定义用户界面。

4) 如何为老年人生成自适应软件用户界面原型?

第 1 步中获取的需求通过第 2 步中的领域特定语言进行建模，然后将这些模型输入第 3 步中的模型驱动工程流程，以生成用户界面原型。这些原型能够在运行时和设计时进行用户界面适配，以满足不同老年最终用户的需求。图 1 展示了一个无障碍场景示例。

5) 如何与老年人一起评估原型?

需要开展一项**用户研究**，以评估老年用户作为**研究参与者**对上一步生成的原型的用户体验。

6）如何利用评估反馈?

根据这些用户研究的反馈，我们将改进目标 1、2、3 和 4 中的制品。总体而言，我们采用**迭代方法**来微调无障碍自适应用户界面生成工具，如图 2 所示。

VI. 结论

全球人口正迅速老龄化，但软件应用程序往往忽视老年人的无障碍需求，从而损害老年用户的用户体验，并进一步导致数字排斥。作为一种解决方案，研究者提出了基于模型驱动工程的用户界面适应性；现有文献中也包含成熟的方法，使开发人员能够在设计时对无障碍使用情境场景进行建模，并指定用户界面适配规则。在运行时，可根据情境变化以自动方式或用户可配置的方式适配用户界面。然而，这些现有工作在将无障碍性整合到其方法和工具中时缺乏全面性，因此我们认为，有必要在用户界面建模阶段将无障碍性视为首要考量。我们的路线图旨在以 WCAG 为框架，设计能够对使用情境场景和用户界面适配规则进行建模的领域特定语言，从而针对任何给定的无障碍场景，在运行时和设计时均可适配用户界面，同时满足网页内容无障碍指南规范。

致谢

作者获 ARC 桂冠研究员奖学金 FL190100035 的支持。

参考文献

[1] 世界卫生组织，“老龄化与健康”，2022年10月，最后访问于2022年12月21日。 [在线]。网址：https://www.who.int/news-room/fact-sheets/detail/ageing-and-health [2] J.-M. D´ıaz-Bossini 和 L. Moreno，“面向老年人的移动界面无障碍性”，Procedia Computer Science，第27卷，第57–66页，2014年。[3] C. Dodd、R. Athauda 和 M. Adam，“面向老年人的用户界面设计：系统性文献综述”，澳大利亚信息系统会议，2017年。[4] L. Elguera Paez 和 C. Zapata Del R´ıo，“老年用户及其在移动应用程序可用性方面面临的主要挑战：系统性综述”，载于人机交互国际会议。Springer，2019年，第423–438页。[5] B. Yousra、L. Zouhaier 和 L. Ben Ayed，“一种根据无障碍使用情境调整用户界面的模型驱动方法：视障用户案例”，Journal on Multimodal User Interfaces，第13卷，2018年11月。[6] P. A. Akiki、A. K. Bandara 和 Y. Yu，“自适应模型驱动用户界面开发系统”，ACM Computing Surveys (CSUR)，第47卷，第1期，第1–33页，2014年。[7] M. Brambilla、J. Cabot 和 M. Wimmer，《实践中的模型驱动软件工程：第二版》，第2版。Springer Cham，2017年5月。[8] N. Ghaibi、O. Dˆaassi 和 L. J. B. Ayed，“基于业务规则管理系统的用户界面适配工具支持”，载于第29届澳大利亚计算机与人类交互会议论文集，系列：OZCHI ’17。美国纽约州纽约：计算机协会，2017年，第162–169页。 [在线]。网址：https://doi.org/10 1145 3152771 3152789 . / . P. Akiki、A. [9] Bandara 和 Y. Yu，“自适应模型驱动用户界面的工程化”，IEEE Transactions on Software Engineering，第42卷，第12期，第1118–1147页，2016年12月。[10] E. Yigitbas、I. Jovanovikj、K. Biermeier、S. Sauer 和 G. Engels，“自适应用户界面的集成模型驱动开发”，Soft ware and Systems Modeling，第19卷，第5期，第1–25页，2020年9月。[11] W3C Web 无障碍倡议（WAI），“《网页内容无障碍指南》（WCAG）2.1”，2018年6月，最后访问于2022年12月21日。[在线]。网址：https://www.w3.org/TR/WCAG21/ [12] C. Stephanidis，《面向所有人的用户界面：概念、方法与工具》。CRC Press，2000年。[13] G. Calvary、J. Coutaz、L. Bouillon、M. Florins、Q. Limbourg、L. Marucci、F. Paterno、C. Santoro、N. Souchon、D. Thevenin 等，“Cameleon 参考框架”，交付成果 D1，第1卷，2002年。[14] A. Braham、M. Khemaja、F. Buend´ıa 和 F. Gargouri，“迈向用于生成自适应用户界面的模型驱动、本体驱动架构”，载于《环境智能——软件与应用——第12届环境智能国际研讨会》，P. Novais、J. Carneiro 和 P. Chamoso 编。Springer International Publishing，2022年，第139–148页。[15] F. Bacha、K. Oliveira 和 M. Abed，“一种专注于内容个性化的用户界面生成模型驱动架构方法”，载于2011年第五届信息科学研究挑战国际会议，2011年，第1–6页。[16] K. Park 和 S.-W. Lee，“自适应用户界面需求工程的基于模型的方法”，载于《大数据时代的需求工程》，L. Liu 和 M. Aoyama 编。德国柏林、海德堡：Springer Berlin Heidelberg，2015年，第18–32页。[17] M. Brambilla 和 P. Fraternali，《交互流建模语言：使用 IFML 对 Web 和移动应用进行模型驱动的用户界面工程》，第1版。美国加利福尼亚州旧金山：Morgan Kaufmann Publishers Inc.，2014年。[18] D. F. Ferraiolo、R. Sandhu、S. Gavrila、D. R. Kuhn 和 R. Chandramouli，“基于角色的访问控制 NIST 标准提案”，ACM Trans. Inf. Syst. Secur.，第4卷，第3期，第224–274页，2001年8月。[19] P. Akiki，“Chain：为自适应用户界面开发模型驱动的上下文帮助”，Journal of Systems and Software，第135卷，190，2017年10月，165页。[20]S. Bongartz、Y. Jin、F. Patern`o、J. Rett、C. Santoro 和 L. D. Spano，“借助基于模型的语言为智能环境构建自适应用户界面”，载于《环境智能》，F. Patern`o、B. de Ruyter、P. Markopoulos、C. Santoro、E. van Loenen 和 K. Luyten 编。德国柏林、海德堡：Springer Berlin Heidelberg，2012年，第33–48页。

[21] F. Patern`o、C. Santoro 和 L. Spano，“MARIA：一种面向普适环境中面向服务的应用的通用声明式多抽象层语言”，*ACMTrans.Comput.-Hum.Interact.*，第 16 卷，第 4 期，2009 年 11 月。[22] R. Mi˜n´on、F. Patern`o、M. Arrue 和 J. Abascal，“在基于模型的用户界面开发过程中为有特殊需求的人群集成适配规则”，《信息社会中的普遍访问》，第 15 卷，第 1 期，第 153–168 页，2015 年 1 月。