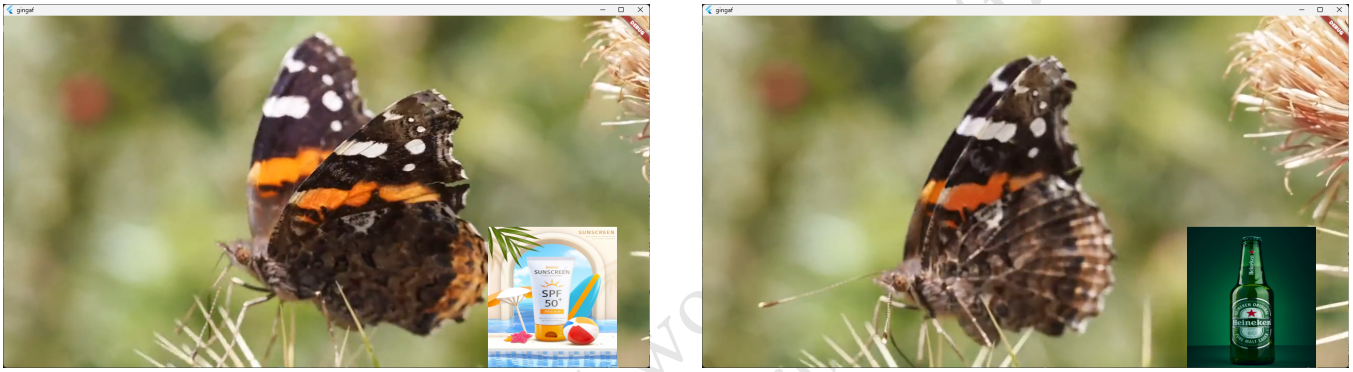
NCL Users API: gingaf support and new proposed user attribute for <rule>

Alan L. V. Guedes
a.guedes@reading.ac.uk
University of Reading
UK

Paulo Victor Borges
paulo.victor.lima15@gmail.com
PUC-Rio
Brazil

Daniel Moraes
danielmoraes@telemidia.puc-rio.br
PUC-Rio
Brazil

Sergio Colcher
colcher@puc-rio.br
PUC-Rio
Brazil



(a) Scenario without adult male user (displays general advertisement).

(b) Scenario with adult male user (displays male advertisement).

Figure 1: NCL application running in two multi-user scenarios.

ABSTRACT

The SBTVD TV 3.0 specification proposes a paradigm shift for its middleware, Ginga, from traditional isolated single-viewer set-top box experiences toward multi-user personalized interactive environments. In this paper, we present the implementation of the **NCL Users API** in `gingaf`, an open-source cross-platform Ginga implementation. We demonstrate how `gingaf` handles dynamic user profile resolution, user settings property binding, and profile data initialization. Furthermore, we propose an extension to the NCL Users API to address the verbosity of causal link assessments by extending NCL's `<rule>` element with a `user` attribute. This attribute allows conditional presentation using `<switch>` elements. Finally, we verify compliance and validate our implementation in multi-user content scenarios.

KEYWORDS

Digital Television, TV 3.0, Ginga, NCL

Unpublished working draft. Not for distribution.

© 2026 SBC – Sociedade Brasileira de Computação.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Anais Estendidos do XXXII Simpósio Brasileiro de Sistemas Multimídia e Web (WTVDI 2026)*, https://doi.org/10.5753/webmedia_estendido.2026.34167.

1 INTRODUCTION

Digital television systems worldwide are undergoing a fundamental transformation toward TV 3.0—a next-generation architecture characterized by advanced audio-visual compression, IP-based broadcast-broadband convergence, multi-screen companion device integration, and personalized user experiences [3]. In modern domestic environments, multiple family members frequently watch shared main-screen broadcasts while simultaneously interacting with personal companion devices such as smartphones, tablets, or secondary displays [7]. Traditional interactive middleware architectures, including early Ginga-NCL specifications [5, 6], evaluated application execution context through global system variables (e.g., `system.language` or `system.caption`). While suitable for single-user set-top box setups, global state models fail in concurrent multi-viewer living room scenarios where distinct individuals require personalized accessibility options, localized language tracks, customized focus navigation, or targeted commercial content on secondary screens.

To overcome these structural limitations, the Brazilian Digital Terrestrial Television (SBTVD) Forum standardized the TV 3.0 specification [2], establishing the *NCL Users API*. This standard defines explicit primitives for managing concurrent user profiles and binding causal hypermedia links to dynamic user properties. This paper introduces an open-source implementation of the NCL Users API in the cross-platform Ginga implementation `gingaf` [4]. Built upon

the headless logic package ncl_doc and Flutter presentation layers, gingaf delivers centralized profile management, reactive property evaluation, and multi-device focus routing.

To support reproducibility, the source code, unit tests, and demonstration videos are available at **https://github.com/ginga-org-br/gingaf**.

The remainder of this paper is structured as follows. Section 2 provides a background on the TV 3.0 Multiuser API. Section 3 describes the design and implementation of multi-user support in gingaf. Section 4 discusses the proposed user attribute for <rule>. Finally, Section 5 presents our final remarks and outlines future work.

2 BACKGROUND ON TV 3.0 MULTIUSER API

Prior to TV 3.0, user settings were represented as key-value pairs stored in the global system node, meaning that modifying a global attribute affected all connected viewers unconditionally. To address these multi-user living room contexts, the TV 3.0 standard [2] introduces new XML elements into the NCL schema to support multi-user profiles and dynamic settings.

First, the **<userProfile>** element, grouped within the <userBase> container (line 21 of Listing 1), represents individual user profiles. It supports local attribute initialization or remote profile queries via the src attribute, as shown on line 22 of Listing 1. The src attribute allows dynamic resolution of user profile data from external JSON endpoints, as shown in Listing 2.

Second, a **<media type="application/x-ncl-user-settings">** element (or a settings node via the <settings> element) binds these user profile attributes to NCL variables. As shown on line 26 of Listing 1, it maps attributes using the pAdult alias, which acts as a dynamic reference pointing to the active viewer interacting with the application.

Third, causal connectors defined in the <connectorBase> (line 9 of Listing 1) evaluate user properties using assessment statements. The onBeginTestVarStart connector (line 10) tests if a node property variable equals a parameter value $value upon element start. As shown on lines 35 and 41, two causal links bind the gender property of uSettings and the area anchor at 2 seconds (aSec2, line 31) of mVideo: the first evaluates if gender equals "male" to start mMaleAd, while the second evaluates if gender equals "female" to start mGeneralAd.

This enables single-device broadcast receivers to evaluate targeted advertisement links (e.g., selecting male-targeted commercial breaks when gender = "male") synchronously upon document initialization. The Users singleton maintains the registry of active users. When a query is run against these users, the system parses the query schema from the profile source, as defined in Listing 2.

```
1  <ncl>
2  <head>
3  <regionBase>
4    <region id="rgAd" left="75%" top="75%" width="20%" height="20%"/>
5  </regionBase>
6  <descriptorBase>
7    <descriptor id="dAd" region="rgAd"/>
8  </descriptorBase>
9  <connectorBase>
10   <causalConnector id="onBeginTestVarStart">
11     <compoundCondition operator="and">
12       <simpleCondition role="onBegin"/>
13       <assessmentStatement comparator="eq">
14         <attributeAssessment role="var" attributeType="nodeProperty"
           eventType="attribution"/>
15         <valueAssessment value="$value"/>
16       </assessmentStatement>
17     </compoundCondition>
18     <simpleAction role="start"/>
19   </causalConnector>
20 </connectorBase>
21 <userBase>
22   <userProfile id="pAdult" src="adult_query.json" max="1"/>
23 </userBase>
24 </head>
25 <body>
26   <settings id="uSettings" type="application/x-ncl-user-settings"
       user="pAdult">
27     <property name="gender"/>
28   </settings>
29   <port id="pMain" component="mVideo"/>
30   <media id="mVideo" src="video.mp4">
31     <area id="aSec2" begin="2s"/>
32   </media>
33   <media id="mMaleAd" src="ad_male.png" descriptor="dAd"/>
34   <media id="mGeneralAd" src="ad_general.png" descriptor="dAd"/>
35   <link xconnector="onBeginTestVarStart">
36     <bind role="onBegin" component="mVideo" interface="aSec2"/>
37     <bind role="var" component="uSettings" interface="gender"/>
38     <bindParam name="value" value="male"/>
39     <bind role="start" component="mMaleAd"/>
40   </link>
41   <link xconnector="onBeginTestVarStart">
42     <bind role="onBegin" component="mVideo" interface="aSec2"/>
43     <bind role="var" component="uSettings" interface="gender"/>
44     <bindParam name="value" value="female"/>
45     <bind role="start" component="mGeneralAd"/>
46   </link>
47 </body>
48 </ncl>
```

Listing 1: video.ncl source code

Listing 2: adult_query.json

```
{"attribute": "age", "comparator": "gte", "value": "18"}
```

3 GINGAF MULTI-USER SUPPORT

gingaf is a decoupled open-source digital television middleware player executing NCL applications via the headless package ncl_doc and a Flutter rendering engine. To illustrate this behavior, we describe two scenarios running the previously detailed NCL application.

gingaf saves user attributes in NCLUserData, which maintains a key-value map for variables. Initial user profiles can be injected into gingaf via the usersDataSrc configuration parameter, which should contain the path to a JSON file with a list of users and their properties. Algorithm 1 details the parsing and profile matching process given an NCL document D and user dataset $Users$.

To illustrate multi-user support in gingaf, we execute the application from Listing 1 for two multi-user scenarios.

The first is a **scenario without a male user** (Figure 1a), using the user profile definitions described in Listing 3. It has two female users: one adult and one underage.

Listing 3: users_data_no_adult_male.json

```
[
  {"id": "u2", "name": "Alice", "gender": "female", "age": 30},
  {"id": "u3", "name": "Kid", "gender": "female", "age": 10}
]
```

Unpublished working draft. Not for distribution.

Algorithm 1 User Profile Properties Parsing

input: NCLDocument D, usersData $Users$

1: **for** each <media> $m \in D$ of type application/x-ncl-user-settings **do**
2: $p \leftarrow$ loadUserProfileQuery(m.user)
3: $u \leftarrow$ matchProfile($Users, p$)
4: m.loadPropertiesFromUser(u)
5: **end for**
6: Execute NCLDocument D

The second is a **scenario with a male user** (Figure 1b), using the user profile definitions described in Listing 4. It has one adult male user and one underage female user. In this case, the adult profile query resolves to the male user (Bob), triggering the targeted advertisement.

Listing 4: users_data_with_adult_male.json

```
[
  {"id": "u1", "name": "Bob", "gender": "male", "age": 30},
  {"id": "u3", "name": "Kid", "gender": "female", "age": 10}
]
```

4 PROPOSED USER ATTRIBUTE FOR <RULE>

To avoid the verbosity of explicit <valueAssessment> statements within causal links, NCL authors can alternatively handle conditional media presentation using the <switch> element. However, as specified in Section 7.2.9 (Presentation control functionality) of ABNT NBR 15606-2:2023 [1], <rule> elements defined within the <ruleBase> implicitly target properties from global system variables (e.g., system.language):

```
<ruleBase>
  <rule id="en" var="system.language" value="en" comparator="eq"/>
</ruleBase>
```

Listing 5: Standard <rule> definition targeting global system variables.

To support multi-user living room contexts without requiring verbose link assessments, we propose extending the <rule> element with an explicit user attribute (e.g., user="pAdult"). This parameter specifies which user settings node or profile is being evaluated during rule resolution. Listing 6 demonstrates this proposed syntax within an NCL application <switch>.

```
<ncl>
<head>
<regionBase>
  <region id="rgAd" left="75%" top="75%" width="20%" height="20%"/>
</regionBase>
<descriptorBase>
  <descriptor id="dAd" region="rgAd"/>
</descriptorBase>
<userBase>
  <userProfile id="pAdult" src="adult_query.json" max="1"/>
</userBase>
<ruleBase>
  <rule id="rMale" user="pAdult" var="gender" comparator="eq" value="male"/>
</ruleBase>
</head>
<body>
  <settings id="uSettings" type="application/x-ncl-user-settings" user="pAdult">
    <property name="gender"/>
  </settings>
  <port id="pMain" component="swAd"/>
  <switch id="swAd">
    <bindRule rule="rMale" constituent="mMaleAd"/>
    <defaultComponent component="mGeneralAd"/>
    <media id="mMaleAd" src="ad_male.png" descriptor="dAd"/>
    <media id="mGeneralAd" src="ad_general.png" descriptor="dAd"/>
  </switch>
</body>
</ncl>
```

Listing 6: Proposed NCL application using <switch> and user parameter at <rule>.

5 FINAL REMARKS

We presented gingaf, a complete open-source implementation of the NCL Users API conforming to TV 3.0 specifications. Future work will expand gingaf to incorporate user profile authentication with companion devices.

REFERENCES

[1] ABNT. 2023. *ABNT NBR 15606-2: Digital Terrestrial Television – Ginga-NCL for Fixed and Mobile Receivers – XML Application Language for Application Coding.* Technical Report. Associação Brasileira de Normas Técnicas.

[2] ABNT. 2024. *ABNT NBR 25608: Digital Terrestrial Television – TV 3.0 Application Coding Specification.* Technical Report. Associação Brasileira de Normas Técnicas.

[3] Joel dos Santos, Li-Chang Shuen Cristina Silva Sousa, Carlos Soares Neto, Guido Lemos, Sérgio Colcher, Marcelo F. Moreno, and Débora Muchaluat-Saade. 2026. An Overview of the TV 3.0 Application Coding Layer Specification. *IEEE Transactions on Broadcasting* 72, 1 (2026), 1–15. doi:10.1109/TBC.2026.3689351

[4] Alan Guedes. 2026. gingaf: Headless and Multiplatform Execution of ITU-SBTVD NCL Applications. In *Proceedings of the 34th ACM International Conference on Multimedia* (Rio de Janeiro, Brazil) *(MM '26).* doi:10.1145/3767308.3834744

[5] ITU-T. 2009. *H.761: Nested Context Language (NCL) and Ginga-NCL for IPTV Services.* Technical Report. International Telecommunication Union, Geneva, Switzerland.

[6] Luiz Fernando Gomes Soares and Rogério Ferreira Rodrigues. 2010. Nested Context Language 3.0: Part 1 – NCL Language. In *Monografias em Ciência da Computação.* PUC-Rio.

[7] Gabriel Souza, D. Silva, M. Delgado, and R. Rodrigues. 2020. Interactive 360-Degree Videos in Ginga-NCL Using Head-Mounted Displays as Second Screen Devices. In *Proceedings of the 26th Brazilian Symposium on Multimedia and the Web.* 113–120. doi:10.1145/3428658.3430972

Unpublished working draft. Not for distribution.