

Программные решения для динамического изменения пользовательского интерфейса на основе автоматически собранной информации о пользователе

В.В. Зосимов <zosimovvv@gmail.com>

А.В. Христоводоров <belfegor26@gmail.com>

А.С. Булгакова <sashabulgakova2@gmail.com>

*Николаевский национальный университет им. В.А. Сухомлинского
Украина, 54030, г. Николаев, ул. Никольская, 24*

Аннотация. В статье описываются функциональные возможности и структура программного модуля для автоматизированной адаптации интерфейсов веб-приложений. Особое внимание уделяется идентификации и различению псевдоанонимных пользователей веб-приложений для адаптации интерфейса под конкретного пользователя. Разработанный подход обеспечивает возможность псевдоидентификации киберсущностей в контексте поведения пользователей и автоматизированную адаптацию интерфейсов под особенности выделенного пользователя в зависимости от поставленных задач.

Ключевые слова: интерфейс; адаптивность; индуктивное моделирование; веб-интерфейс

DOI: 10.15514/ISPRAS-2018-30(3)-15

Для цитирования: Зосимов В.В., Христоводоров А.В., Булгакова А.С. Программные решения для динамического изменения пользовательского интерфейса на основе автоматически собранной информации о пользователе. Труды ИСП РАН, том 30, вып. 3, 2018 г., стр. 207-220. DOI: 10.15514/ISPRAS-2018-30(3)-15

1. Введение

Интерфейсы – это неотъемлемая составляющая звеньев восприятия программных продуктов и управления ими. Именно они обеспечивают управление программными продуктами и связь пользователей с программой. Неустанная интеграция IT-технологий в среду существования современного человека порождает все больший спрос на разработки в области адаптации

интерфейсов программных продуктов, направленных на удовлетворение потребностей пользователей.

Программные продукты, ориентированные на повышение удобства пользования интерфейсами, обладают следующими достоинствами:

- упрощение восприятия бизнес-логики программного продукта конкретным пользователем;
- гибкость модели представления информации;
- повышение производительности работы с интерфейсом.

Под динамическим изменением интерфейса следует понимать его адаптацию в результате выполнения некоторого сценария на основе поведенческого портрета пользователя.

Наиболее полный обзор содержания адаптивного поведения человеко-машинного интерфейса (ЧМИ) содержится в [1]. Авторы предлагают три параметра интерфейса, которые могут меняться:

- содержание представляемой информации;
- форма представления информации и ведения диалога;
- распределение задач между человеком и машиной (уровень автоматизации).

В других работах декларируется, что адаптивность интерфейса проявляется:

- в настройке уровня детализации диалога с пользователем – от подробного диалога, «ведущего» пользователя к цели шаг за шагом через иерархию меню, к короткому, с использованием сокращенных команд и макросов в режиме «вопрос-ответ» [2];
- в подсказках, ограничении доступа к приложениям, регулировании интенсивности информационного обмена и изменению внешнего вида интерфейса [3,4];
- в фильтрации и расстановке приоритетов контента, предложенного пользователю; это может происходить не только в соответствии с собственными предпочтениями, но и под влиянием внешних факторов и контекста (например, пользователю предлагается продукт, который выгодно купить именно сейчас [5]);
- в изменении темпа подачи информации [6];
- в настройке параметров изображения (толщины линий, размера шрифта, яркости и др.).

В условиях конкуренции современных разработок в области разработки и применения интерактивных интерфейсов повышается актуальность исследований в направлении автоматизированной адаптации пользовательских интерфейсов. Под автоматизированной адаптацией понимается динамическое изменение пользовательских интерфейсов на основе автоматически собранной информации о пользователе.

Значительную роль в исследовании моделей и методов адаптации интерфейсов играют достижения Питера Экерсли [7]. Отдельные вопросы проектирования и анализа использования возможностей псевдоидентификации рассматриваются в трудах исследователей Г. Кришна, К. Докинз, Дж. Гилмор [8, 9]. Некоторые вопросы использования технологий сбора данных в автономных киберфизических системах рассматриваются в работах отечественных ученых А. Бочкарева, В. Голембо [10].

В [8-10] рассмотрены фундаментальные принципы и концепции, которые применяются при реализации программных продуктов, представлены различные модели и методы, направленные на улучшение восприятия и взаимодействия с интерфейсами конечных программных продуктов. Но все эти аспекты рассматривались как отдельные задачи, и отсутствовало их объединение в единую адаптивную систему.

Анализ результатов предыдущих исследований позволил прийти к выводу, что на сегодняшний день неуклонно растет спрос на системы адаптации интерфейсов во всех сферах их использования, а работоспособные системы отсутствуют. В данной статье описывается разработанный авторами подход, доведенный до программной реализации.

1. Индуктивный подход к построению адаптивных интерфейсов

Пусть $CR = \{cr_1, \dots, cr_{|CR|}\}$ – множество критериев, которые будут изменяться, то есть адаптироваться под пользователя, $Fact = \{fact_1, \dots, fact_{|Fact|}\}$ – множество факторов, которые будут влиять на выбор того или иного критерия.

На основе выделенных заранее факторов, которые могут влиять на интерфейс, будут меняться критерии построения интерфейса. Индуктивность процесса заключается в том, что адаптация интерфейса происходит от конкретных данных наблюдений, то есть факторов – к общей модели [11], которая включает в себя множество критериев, которые будут меняться (например, при известной информации о возрасте пользователя (фактор – возраст), увеличить размер шрифта (для пожилых людей) – критерий).

Следовательно, исходя из вышесказанного, можно построить функцию Φ , отражающую процесс адаптации, где θ – параметры модели:

$$\begin{aligned} \Phi(cr_1(fact_1, \dots, fact_{\tilde{\kappa}}), \dots, cr_m(fact_{\tilde{\kappa}}, \dots, fact_F)) = & \theta_0 + \sum_{i=1}^m \theta_i cr_i \sum_{F=1}^{\tilde{\kappa}} (fact_1, \dots, fact_F) + \\ & + \sum_{i=1}^m \sum_{j=1}^m \theta_{ij} cr_i cr_j \sum_{F=1}^{\tilde{\kappa}} (fact_1, \dots, fact_F) + \sum_{i=1}^m \sum_{j=1}^m \sum_{\tilde{\kappa}=1}^m \theta_{ijk} cr_i cr_j cr_{\tilde{\kappa}} \sum_{F=1}^{\tilde{\kappa}} (fact_1, \dots, fact_F) + \dots \end{aligned}$$

Общая функциональная схема механизма адаптации интерфейса представлена на рис.1. Механизм состоит из нескольких блоков: блок определения

факторов, где формируются/хранятся факторы пользователя, которые могут влиять на критерии изменения интерфейса, то есть адаптации; блок формирования критериев (количество критериев для каждого фактора может быть разной); блок обработки данных, в котором программным образом будет обрабатываться содержание критериев, приводит к изменению интерфейса [12].

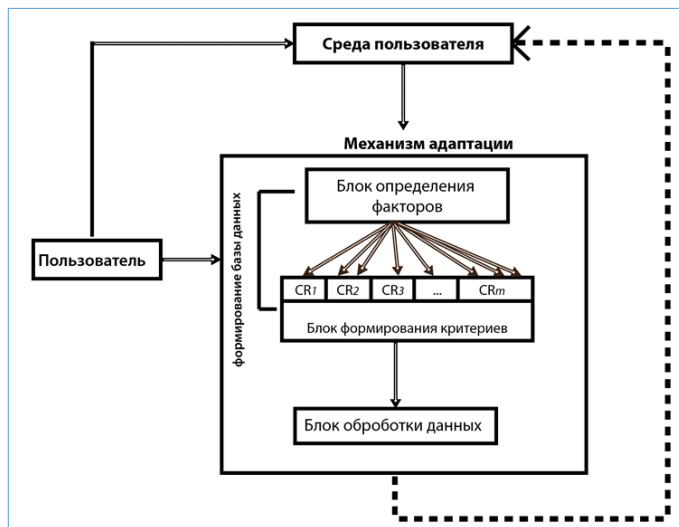


Рис.1. Составные части механизма адаптации интерфейса
Fig.1. Components of the interface adaptation mechanism

2. Требования к программному продукту

Программный продукт должен обеспечивать следующие функциональные возможности:

- сбор и хранение информации о пользователях соответствующего веб-приложения;
- псевдоидентификацию пользователей на основании собранных данных;
- автоматизированную адаптацию интерфейсов псевдоидентифицированных пользователей.

Программный продукт должен соответствовать следующим требованиям:

1. контролируемый сбор информации о пользователях веб-приложения;
2. создание и поддержка базы данных полученной информации;
3. псевдоидентификация пользователей веб-приложения на основе собранной информации;

4. автоматизированная адаптация пользовательского интерфейса веб-приложения на основании собранных и обработанных программным продуктом данных.

Ниже проведен сравнение предъявленных требований с возможностями существующих аналогов. За последние годы сформировалось и продолжает развиваться направление корректировки контента, структуры сайта, контроля внутренних и внешних факторов пребывания и взаимодействия пользователя с интерфейсом. Это неотъемлемая составляющая успеха маркетинговых и информационных сайтов. Существует достаточно большое количество систем сбора информации о пользователях, например, Amplitude [13], Mixpanel [14] и др. Эти программные продукты удовлетворяют требованиям 1 и 2, но не удовлетворяют требованиям 3 и 4. Практически отсутствуют системы, которые удовлетворяют всем четырем перечисленным требованиям.

3. Структурная схема программного продукта

На рис. 2 представлена схема взаимодействия программной системы «AAUI - Automatic Adaptation of User Interfaces».

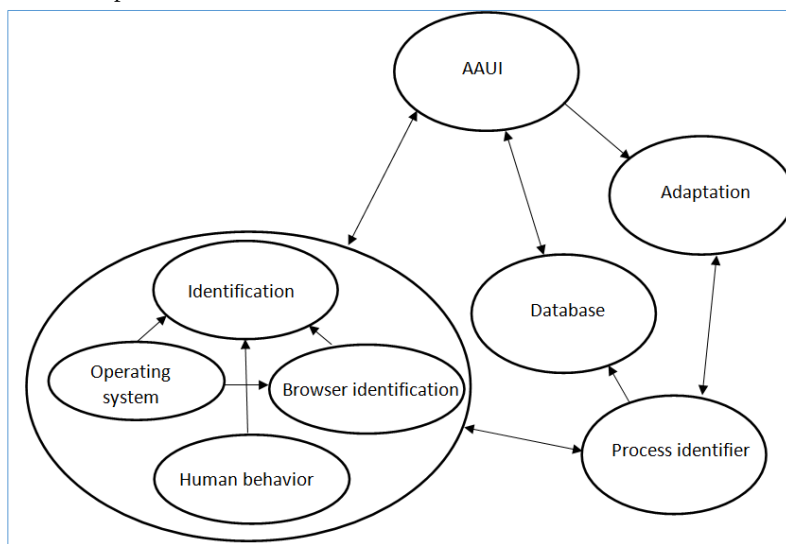


Рис 2. Схема взаимодействия программной системы «AAUI»

Fig. 2. Diagram of interaction of the "AAUI" software system

Блок «AAUI» является начальным модулем, начиная с которого происходит взаимодействие с компонентами программного продукта. «Identification» реализует формирование сущности на основании собранной информации об объекте, который проявил активность. «Operating system» обеспечивает сбор

информации о программных средствах активного объекта посещения. «Browser identification» отвечает за сбор информации о программном обеспечении, которое задействовано во взаимодействии с конечным программным продуктом. «Human behavior» обеспечивает проверку возможного прошлого присутствия псевдодеанимизированного объекта. «Adaptation» отвечает за программную адаптацию на основе созданного отпечатка псевдодеанимизированного объекта взаимодействия с конечным продуктом. «Database» отвечает за сохранность проанализированной информации, токенов псевдодеанимизации и правил адаптации интерфейсов. «Process identifier» отвечает за согласование идентификации существующих объектов взаимодействия.

На рис. 3 представлена блок-схема использования программного продукта.

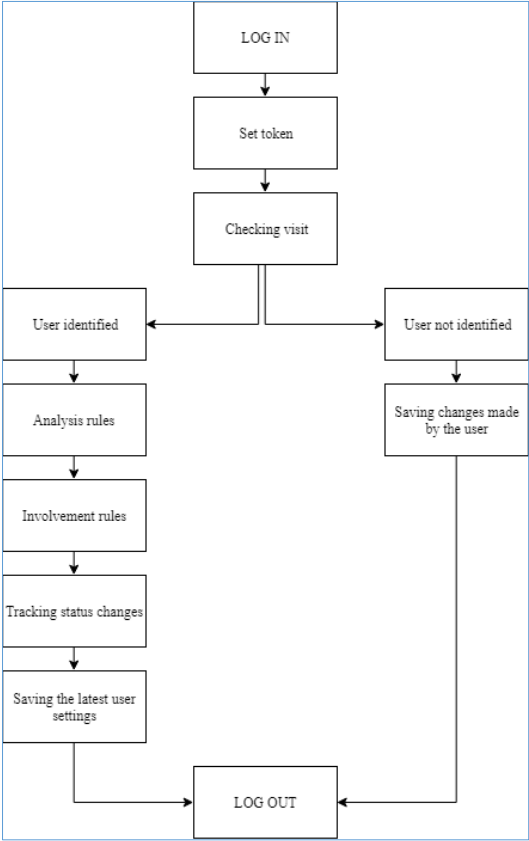


Рис 3. Схема использования программного продукта
Fig. 3. Scheme of using the software

В блоке «Set token» – программная система «AAUI» генерирует идентификатор. Генерация идентификатора происходит параллельно со сбором дополнительных данных для более точной идентификации пользователя, рис. 4.

```
171 function _getBrowserFingerprint() {  
172     $useragent = $_SERVER['HTTP_USER_AGENT'];  
173     $accept    = $_SERVER['HTTP_ACCEPT'];  
174     $charset   = $_SERVER['HTTP_ACCEPT_CHARSET'];  
175     $encoding  = $_SERVER['HTTP_ACCEPT_ENCODING'];  
176     $language  = $_SERVER['HTTP_ACCEPT_LANGUAGE'];  
177     $data = '';  
178     $data .= $useragent;  
179     $data .= $accept;  
180     $data .= $charset;  
181     $data .= $encoding;  
182     $data .= $language;  
183  
184     $hash = hash('sha256', $data);  
185     return $hash;  
186 }
```

Рис 4. Реализация блока генерации идентификатора

Fig. 4. Implementation of the ID generation unit

«Checking visit» – проверка на возможного предыдущего посещения сервиса пользователем.

- Если пользователь идентифицируется («User identified») происходит анализ поведенческих характеристик «Analysis rules», собранных на основе объектов интерфейса программного продукта, ориентированных на отслеживание изменений; применение персональных настроек «Involvement rules» и дальнейшее наблюдение за изменением потенциально изменяемых настроек интерфейса «Tracking status changes». Все изменения настроек веб-интерфейса, которые отслеживаются, сохраняются в базе данных каждый раз, когда они происходят – «Saving the latest user settings»;
- В случае, если пользователь впервые посещает веб-сервис, т.е. «User not identified», происходит отслеживание изменений, задаваемых пользователем, и их в базе данных сохранение «Saving changes made by the user».

4. Программная реализация

Основным преимуществом программной системы является примененная модель идентификации анонимных пользователей конечных программных продуктов с дальнейшим использованием динамического идентификатора для автоматической адаптации интерфейса под идентифицированного пользователя.

«AAUI» является серверным программным продуктом, написанным на скриптовом языке программирования PHP. Программный продукт обеспечивает:

- администрирование программного продукта (управление списком задействованных фильтров идентификации, изменение, редактирование, создание правил адаптации);
- хранение и обработка информации о пользователях;
- обеспечение хранения информации о сессиях.

В «AAUI» предусмотрен механизм настройки псевдодеанонимизации пользователей, и их группировки по заданным фильтрам. Функции групп пользователей:

1. группировка списков по странам посещения;
2. группировка списков пользователей по языку пользователей;
3. группировка по конфигурации конечного устройства посетителя;
4. комбинация групп для выделения целевой аудитории.

Надежность и устойчивое функционирование «AAUI» достигается совокупностью следующих организационно-технических мероприятий:

1. организация надежной защиты специалистами по кибербезопасности веб-приложений;
2. организации регулярного и качественного технического обслуживания серверной части;
3. своевременное обслуживание базы данных.

Серверное приложение «AAUI» взаимодействует с посетителем веб-приложения с начала его взаимодействия с интерфейсом веб-приложения (рис. 5). Уровень «поглощения» собранной информации о посетителе устанавливает администратор программной системы.



Рис 5. Схема модели взаимодействия
Fig. 5. Diagram of interaction model

Основным источником хранения данных является база данных. В ней хранится информация о полученных сущностях псевдодеанонимизированных пользователях. Для хранения информации используется свободная система управления реляционными базами данных MySQL. Структура базы данных разработана на основе разработанной методологии идентификации пользователей.

Для визуального представления схемы базы данных была использована методология IDEF1X, направленная на моделирование реляционных баз данных (рис.6). Указанный стандарт входит в семейство методологий IDEF позволяющее исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем. Методология IDEF1X адаптирована для совместного использования с IDEF0 в рамках единой технологии моделирования. На основе IDEF0 детализируются функциональные блоки, а IDEF1X позволяет детализировать «стрелки». Разработка базы данных производилась с помощью инструмента для проектирования баз данных MySQL Workbench – инструмента, интегрирующего проектирование, моделирование, создание и эксплуатацию БД в единое окружение для системы баз данных MySQL с использованием, в частности, нотации IDEF1X.

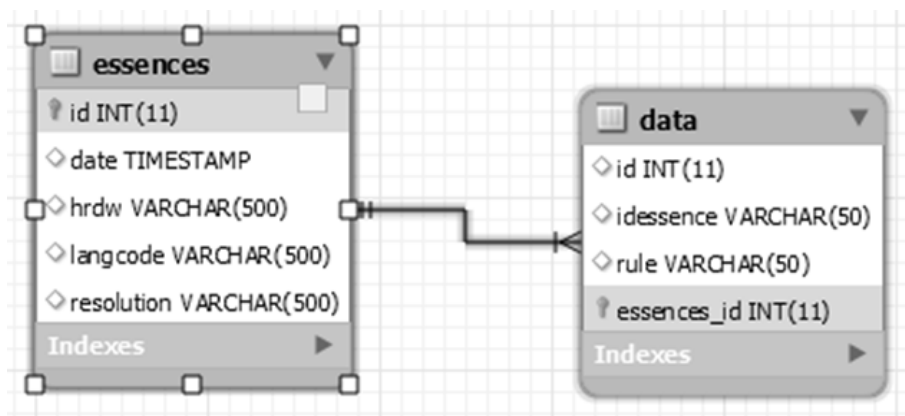


Рис 6. Фрагмент ER-диаграммы базы данных в нотации IDEF1X
Fig. 6. Fragment of the ER-diagram of the database in IDEF1X notation

Интересной в данном продукте является реализация системы псевдоидентификации пользователей. Взаимодействие частей программного продукта демонстрируют части кода, представленные на рис.7 и 8. На рис.7. представлен блок получения информации об установленных пользователем языков и часового пояса. Функция автоматической адаптации локализации веб-приложения на основе собранных данных представлена на рис.8.

5. Экспериментальное использование технологии

После интеграции «AAUI» с Интернет магазином было проведено исследование, направленное на выявление процентного соотношения прироста количества новых посетителей. В течение тридцати дней система работала в режиме сбора поведенческих сущностей, псевдодеанимизации и формирования базы посетителей.

Была получена выборка из 2000 уникальных поведенческих портретов. Из них 30% возвращались на сайт Интернет магазина в дальнейшем. На основании анализа собранных данных была запрограммирована функция адаптации, которая была более всего востребована – сортировка товаров по цене (от минимальной до максимальной).

```
118 App::before(function($request){
119     $url_lang = Request::segment(1);
120     $cookie_lang = Cookie::get('language');
121     $browser_lang = substr(Request::server('HTTP_ACCEPT_LANGUAGE'), 0, 2);
122     if(!empty($url_lang) AND in_array($url_lang, Config::get('app.languages'))))
123     {
124         if($url_lang != $cookie_lang)
125         {
126             Session::put('language', $url_lang);
127         }
128         App::setLocale($url_lang);
129     }
130     else if(!empty($cookie_lang) AND in_array($cookie_lang, Config::get('app.languages'))))
131     {
132         App::setLocale($cookie_lang);
133     }
134
135     else if(!empty($browser_lang) AND in_array($browser_lang, Config::get('app.languages'))))
136     {
137         if($browser_lang != $cookie_lang)
138         {
139             Session::put('language', $browser_lang);
140         }
141         $timezone = \Auth::user()->timezone;
142
143         $datetime = $this->asDateTime($value);
144         DB::table('essences')->insert(
145             ['langcode' => , $browser_lang] 'votes' => 0]
146             ['date' => , $datetime + $timezone] 'votes' => 0]
147         );
148     }
149     else
```

Рис 7. Блок сбора информации
Fig. 7. Information collection block

```
203 public function handle($request, Closure $next)
204 {
205     if(!\Session::has('locale'))
206     {
207         \Session::put('locale', \Config::get('app.locale'));
208     }
209     $usersrule = DB::table('data')->select('rule', 'rule')->get();
210     if(checkrule($usersrule) == Config::get('app.locale'))
211
212     app()->setLocale(\Session::get('usersrule'));
213
214     return $next($request);
215 }
```

Рис 8. Функция автоматической адаптации
Fig. 8. Automatic adaptation function

Программная система «AAUI», в течение тридцати дней генерировала выборку поведенческих портретов. При повторном посещении псевдоеанимизированного пользователя система автоматически адаптировала запрограммированную на отслеживание функцию в зависимости от полученного состояния сортировки.

В результате использования «AAUI» процент посетителей веб-приложения, которые снова обратились в Интернет-магазин, вырос с 30% до 50%.

В дальнейшем будет проведено более детальное экспериментальное исследование возможностей предлагаемой технологии.

6. Заключение

В работе представлено описание программного обеспечения для автоматизированной адаптации пользовательских интерфейсов. Проведенный анализ систем, направленных на сбор информации о пользователях, таких как Amplitude и Mixpanel, позволил сделать вывод, что указанные программные системы обеспечивают только сбор профилированной информации о входящих пользователях и не обеспечивают автоматизированной адаптации интерфейсов под нужды пользователей.

Разработанная система «AAUI» направлена на автоматизированную адаптацию интерфейсов под нужды пользователей. Система обеспечивает псевдоидентификацию пользователей (построение базы анонимных пользователей и правил на основе их пребывания в веб-приложениях). Оригинальной особенностью системы является использованная модель идентификации анонимных пользователей конечных программных продуктов с дальнейшим использованием динамического идентификатора для автоматической адаптации интерфейса под идентифицированного пользователя. В открытом доступе отсутствует аналогичное программное обеспечение.

К перспективам дальнейшей разработки «AAUI» можно отнести увеличение количества маркеров идентификации для повышения достоверности идентификации пользователей, внедрение в системы управления контентом и оптимизацию работы с данными.

Представлено экспериментальное применение технологии на примере Интернет-магазина, в котором в течение 30 дней было зафиксировано 2000 уникальных поведенческих портретов. В результате эксперимента процент посетителей веб-приложения, которые снова обратились в Интернет-магазин, вырос с 30% до 50%.

Список литературы

- [1] Rothrock L., Koubek R., Fuchs F. et al. Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms. *Theoretical Issues in Ergonomics Science*, vol. 3, No. 1, 2002, pp. 47-84.

- [2] Ходаков В.Е., Ходаков Д.В. Адаптивный пользовательский интерфейс: проблемы построения. Автоматика. Автоматизация. Электротехнические комплексы и системы, № 1 (11), 2002, стр. 45-57.
- [3] Курзанцева Л.И. Методика комплексного исследования адаптивного человеко-машинного интерфейса. Математические машины и системы, № 4, 2011, стр. 69-77.
- [4] Курзанцева Л.И. Об адаптивном интеллектуальном интерфейсе «пользователь – система массового применения». Компьютерные средства, сети и системы, №7, 2008, стр. 110-116.
- [5] Langley P. User modeling in adaptive interfaces. In Proceedings of the Seventh International Conference on User Modeling, 1999, pp. 357-370.
- [6] Karwowski W. A review of human factors challenges of complex adaptive systems: discovering and understanding chaos in human performance. Human Factors, vol. 54, No. 6, 2012, pp. 983-995.
- [7] Peter Eckersley, How Unique Is Your Web Browser? Electronic Frontier Foundation Режим доступа: <https://www.eff.org/>, дата обращения 02.05.2017.
- [8] Gilmore J. Easy Laravel 5. Leanpub, 2016, 235 p.
- [9] Dockins K. Design Patterns in PHP and Laravel. Appers, 2017, 45 p.
- [10] О. Бочкарьов, В. Голембо. Використання інтелектуальних технологій збору даних у автономних кіберфізичних системах. Lviv Polytechnic National University Institutional Repository. Сборник научных трудов, № 830, 2015, стр. 7–11 (на украинском).
- [11] Stepashko V., Bulgakova O., Zosimov V. Construction and research of the generalized iterative GMDH algorithm with active neurons. Advances in Intelligent Systems and Computing, vol. 689, 2018, pp. 492-510.
- [12] Булгакова А.С. Концепция построения адаптивного интерфейса с использованием индуктивного подхода. Индуктивное моделирование сложных систем. Сборник трудов, выпуск 8. Киев: МННЦ ИТС, 2016, стр. 73-78 (на украинском)
- [13] Amplitude. Analytics for modern product teams. Режим доступа: <https://amplitude.com/>, дата обращения 02.05.2017.
- [14] Mixpanel. Product and User Analytics for Mobile, Web, and Beyond. Режим доступа: <https://mixpanel.com/>, дата обращения 02.05.2017.

Dynamically changing user interfaces: software solutions based on automatically collected user information

V.V. Zosimov <zosimovvv@gmail.com>
O.V. Khrystodorov <belfegor26@gmail.com>
O.S. Bulgakova <sashabulgakova2@gmail.com>
V.O. Sukhomlynsky Mykolaiv National University,
24 Nikolska St, Mykolayiv, 54030, Ukraine

Abstract. The developed system "AAUI" is aimed at the automated adaptation of interfaces to the needs of users. The system provides pseudo-identification of users (building anonymous users and rules database based on their stay in web applications). The original feature of the system is the used model of identifying anonymous users of the end products

with the further use of the dynamic identifier to automatically adapt the interface to the identified user's need. The analysis of the systems aimed at collecting information about users made it possible to conclude that these software systems provide only the collection of profiled information about incoming users and do not provide for automated adaptation of interfaces to the needs of users. There is no similar software in the open access. The prospects for the further development of AAUI include an increase in the number of identification markers to improve the authenticity of user identification, the integration with content management systems and the optimization of data management.

Keywords: interface, adaptiveness, inductive modelling, web-interface

DOI: 10.15514/ISPRAS-2018-30(3)-15

For citation: Zosimov V.V., Khrystodorov O.V., Bulgakova O.S. Dynamically changing user interfaces: software solutions based on automatically collected user information. *Trudy ISP RAN/Proc. ISP RAS*, vol. 30, issue 3, 2018, pp. 207-220 (in Russian). DOI: 10.15514/ISPRAS-2018-30(3)-15

References

- [1] Rothrock L., Koubek R., Fuchs F. et al. Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms. *Theoretical Issues in Ergonomics Science*, vol. 3, No. 1, 2002, pp. 47-84.
- [2] Khodakov V.E., Khodakov D.V. Adaptive user interface: problems of building. *Avtomatika. Avtomatizacija. Jeлектrotehnicheskie komplekсы i sistemy* [Automation. Automatization. Electrotechnical complexes and systems], № 1 (11), 2002, pp. 45-57 (in Russian).
- [3] Kurgantseva L.I. Methodology of integrated studies of the adaptive human-machine interface. *Matematicheskie mashiny i sistemy* [Mathematical Machines and Systems], № 4, 2011, pp. 69-77 (in Russian).
- [4] Kurgantseva L.I. About the adaptive intellectual interface "The user – system of mass application". *Komp'yuternye sredstva, seti i sistemy* [Computer means, networks and systems], №7, 2008, pp. 110-116 (in Russian).
- [5] Langley P. User modeling in adaptive interfaces. In *Proceedings of the Seventh International Conference on User Modeling*, 1999, pp. 357-370.
- [6] Karwowski W. A review of human factors challenges of complex adaptive systems: discovering and understanding chaos in human performance. *Human Factors*, vol. 54, No. 6, 2012, pp. 983-995.
- [7] Peter Eckersley, How Unique Is Your Web Browser? Electronic Frontier Foundation
Режим доступа: <https://www EFF.org/>, дата обращения 02.05.2017.
- [8] Gilmore J. *Easy Laravel 5*. Leanpub, 2016, 235 p.
- [9] Dockins K. *Design Patterns in PHP and Laravel*. Appers, 2017, 45 p.
- [10] Botchkaryov A., Golembo V. Applying intelligent technologies of data acquisition to autonomous cyber-physical systems. *Lviv Polytechnic National University Institutional Repository*, № 830, 2015, стр. 7–11 (in Ukrainian).
- [11] Stepashko V., Bulgakova O., Zosimov V. Construction and research of the generalized iterative GMDH algorithm with active neurons. *Advances in Intelligent Systems and Computing*, vol. 689, 2018, pp. 492-510.

- [12] Bulgakova O. The concept of constructing an adaptive interface using an inductive approach. *Induktivnoe modelirovanie slozhnyh sistem* [Inductive modeling of complex systems], pp. 73-78 (in Ukrain)
- [13] Amplitude. Analytics for modern product teams. Available at: <https://amplitude.com/>, accessed 02.05.2017.
- [14] Mixpanel. Product and User Analytics for Mobile, Web, and Beyond. Available at: <https://mixpanel.com/>, accessed 02.05.2017.