

УДК 004.54

DOI: 10.30987/article_5bd17b42b1a236.42909295

С.А. Сакулин, А.Н. Алфимцев, Е.А. Типсин

ВЕРИФИКАЦИЯ РАСПРЕДЕЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА НА ОСНОВЕ π -ИСЧИСЛЕНИЯ

Описан метод верификации распределения пользовательского интерфейса, в основе которого лежит математический аппарат π -исчисления. Разработаны правила и процедура распределения элементов интерфейса по отдельным устройствам. Приведено пошаговое описание верификации рас-

пределения интерфейса по разработанным правилам в терминах π -исчисления.

Ключевые слова: верификация, формализация, распределение, распределенный интерфейс, пользовательский интерфейс, π -исчисление, редукция.

S.A. Sakulin, A.N. Alfimtsev, E.A. Tipsin

VERIFICATION OF USER INTERFACE DISTRIBUTION BASED ON π -CALCULUS

A rapid growth of computer device quantity resulted in the emergence of distribute user interfaces which mean a user work with the application by means of some devices simultaneously. During the design of a distributed interface it is necessary every time to develop the logic of interface element distribution. Besides, the complexity of DPI element distribution grows exponentially. In particular, for the distribution of an interface consisting of two elements for two devices it is necessary to consider nine possible options of distribution.

Therefore the problem of the formal method development of interface element distribution verification on separate devices is supposed to be urgent. In this work there is presented a method for the verification of

user interface distribution on the basis of the mathematical apparatus of π -calculus which was adapted for the verification of a distribution process. For the formalization of a distribution process each interface element is compared with attributes and each device – with parameters. On the basis of the introduced attributes and parameters there are formulated ten rules for a distribution in accordance of which a procedure is developed for the distribution of interface elements onto separate devices including six steps for which the developed verification method was used and described in terms of π -calculus.

Key words: verification, formalization, distribution, distributed interface, user interface, π -calculus, reduction.

Введение

Распределенные пользовательские интерфейсы (РПИ) подразумевают работу пользователя с приложением посредством нескольких устройств одновременно, что повышает функциональность, продуктивность и удобство использования приложения [1]. В настоящее время существуют следующие реализации РПИ: система синхронизации смартфона с часами; система управления устройствами с помощью телефона или планшета; РПИ для совместного принятия решений и другие. Для проектирования нового РПИ необходимо всякий раз разрабатывать логику распределения элементов интерфейса. Такая ситуация затрудняет массовое внедрение РПИ ввиду непонимания пользователями логики распределения и нежелания испытывать ког-

нитивную нагрузку, связанную с освоением новых интерфейсов [2], а также относительно большими трудозатратами разработчиков. Кроме того, сложность распределения элементов РПИ растет экспоненциально. В частности, для распределения интерфейса, состоящего из двух элементов, на два устройства необходимо рассмотреть девять возможных вариантов распределения, поскольку один элемент РПИ может быть распределен на одно из двух устройств или сразу на оба.

Элементы РПИ используются в ситуационных центрах. В этих центрах экраны коллективного пользования отображают графики, видеоизображения и другие формы графического представления информации. Главной задачей при проекти-

ровании ситуационных центров является компоновка устройств и определение форм представления информации таким образом, чтобы минимизировать время принятия решения. Распределение информации по экранам центра производит оператор [3; 4] с учетом требований эргономики [5].

Одной из смежных областей с распределенными интерфейсами является полиэкранное отображение данных. Большинство работ в этой области связаны с техническими решениями создания систем

для полиэкранного отображения данных, а также уделяется внимание методам синхронизации видеопотоков в вычислительном кластере для обеспечения визуальной непрерывности изображения на полиэкранной системе.

Ввиду изложенного задача разработки формальных методов верификации распределения элементов интерфейса по отдельным устройствам представляется актуальной.

1. Правила распределения элементов интерфейса

К типовым структурным блокам интерфейса относятся основной блок B_1 , заголовков B_2 , подвал B_3 , левая колонка B_4 и правая колонка B_5 [6]. Рассмотрим в качестве примера распределение упрощенного

интерфейса интернет-магазина (рис. 1). Этот интерфейс содержит структурные блоки B_1, B_2, B_4 , логические блоки L_1, \dots, L_4 и элементы E_1, \dots, E_{18} .

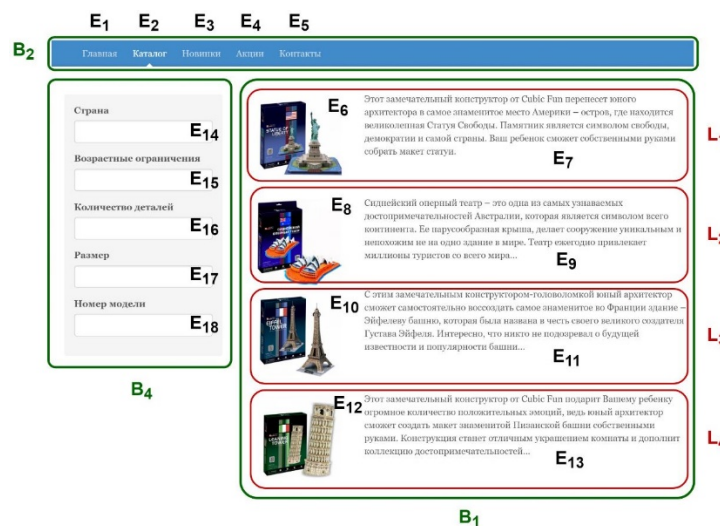


Рис. 1. Упрощенный интерфейс интернет-магазина

Первоначально исходный интерфейс I отображается на устройстве D_1 (рис. 2). Распределенный интерфейс I' располагается на основном $D_1 = D'$ и вспомогательном

D_2 устройствах. Вершинами графа на рис. 2 представлены устройства, структурные блоки, логические блоки, элементы интерфейса.

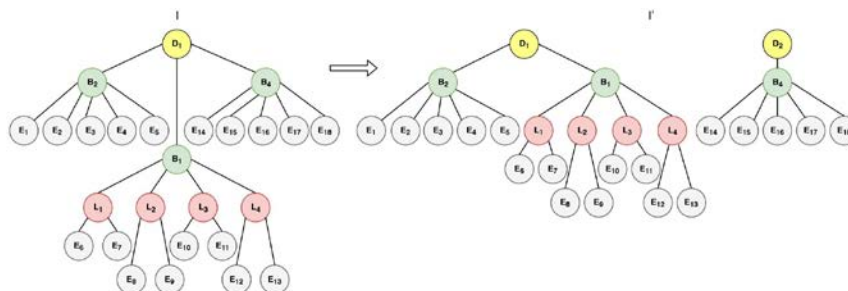


Рис. 2. Распределение исходного интерфейса I

Для формализации логики распределения каждому элементу E интерфейса сопоставим атрибуты a_1^E, \dots, a_{12}^E : a_1^E – идентификатор элемента в виде строки; a_2^E – тип элемента, например изображение, видеоматериал, текст, кнопка и т.д. (представлен строкой с названием типа); a_3^E – размеры элемента в пикселях в виде пары чисел (ширина и высота); a_4^E – координаты центра элемента по горизонтали и вертикали на странице интерфейса (представлены соответствующей парой чисел); a_5^E – бинарный признак активности элемента (1 – активен, 0 – не активен); a_6^E – идентификатор структурного блока, в котором находится элемент, в виде строки; a_7^E – описание логических связей с другими элементами интерфейса в виде множества идентификаторов соответствующих элементов; a_8^E – бинарный признак типа контента (1 – основной контент, 0 – второстепенный контент); a_9^E – атрибут условия распределения (представлен совокупностью пар «параметр устройства, область допустимых значений параметра»); a_{10}^E – атрибут пользовательских предпочтений (представлен совокупностью пар «параметр устройства, область допустимых значений параметра»); a_{11}^E – бинарный признак конфиденциальности элемента (1 – конфиденциальный, 0 – не содержит конфиденциальных данных); a_{12}^E – требования к параметрам устройства, при удовлетворении которых элемент может быть распределен на это устройство (представлены совокупностью пар «параметр устройства, область допустимых значений параметра»).

Каждому устройству D сопоставим параметры c_1^D, \dots, c_5^D : c_1^D – тип устройства в виде строки с его названием; c_2^D – размер диагонали дисплея устройства в виде соответствующего числа; c_3^D – бинарный параметр мобильности (1 – устройство является мобильным, 0 – не является); c_4^D – операционная система устройства в виде

строки с соответствующим названием; c_5^D – бинарный параметр наличия динамиков (1 – устройство воспроизводит звук, 0 – не воспроизводит). Требованиями к устройству C^D назовем совокупность допустимых значений параметров c_1^D, \dots, c_5^D устройства D . Факт удовлетворения устройства D требованиям C^D обозначим как $D \approx C^D$. Факт соответствия требований C^D атрибутам a_1^E, \dots, a_{12}^E элемента E интерфейса обозначим как $C^D \approx \{a_1^E, \dots, a_{12}^E\}$.

Набор устройств $S = \{D_1, \dots, D_n\}$ представляет собой совокупность устройств, на которые будет распределен интерфейс I . Устройство является свободным, если на него не распределен ни один из элементов интерфейса. Факт распределения элемента интерфейса E на устройство D обозначим как $E \rightarrow D$.

Сформулируем правила распределения с использованием введенных обозначений.

1.1. Правило основного устройства основано на том, что пользователи делят все устройства на два типа: устройство для интерфейса основной задачи и остальные устройства для поддержки ее выполнения [7]. В соответствии с этим правилом для интерфейса основной задачи выбирается устройство, которое должно обеспечивать наилучшее взаимодействие пользователя с приложением, т.е. в любом наборе устройств S существует одно основное устройство D' .

1.2. Правило логической связности основано на делении элементов интерфейса на связные логические группы [8]. В соответствии с этим правилом разбиение такой группы при распределении не допускается, т.е. распределение логической группы на какое-либо устройство должно происходить целиком, или, формально: если у какого-либо элемента E атрибут $a_6^E = \{E_1, \dots, E_N\}$, где E_1, \dots, E_N – элементы, входящие в одну логическую группу, то $\{E_1, \dots, E_N\} \rightarrow D$, где D – устройство,

на которое распределяются все эти элементы.

1.3. Правило устоявшихся условностей основано на учете при распределении элементов интерфейса множества сформировавшихся за последние годы устоявшихся условностей [9]. К таким условностям, в частности, относятся: распределение интерфейса управления воспроизведением видеоконтента на вспомогательные мобильные устройства [7; 10]; распределение элементов интерфейса, которые используют геолокацию, на мобильные устройства. Запишем это правило формально: если $a_9^E \neq \emptyset$, $C^D \approx \{a_9^E\}$ и доступно устройство D , такое, что $D \approx C^D$, то $E \rightarrow D$.

1.4. Правило учета пользовательских предпочтений основано на том факте, что пользователи предпочитают некоторые устройства для выполнения определенных задач, несмотря на возникающие при этом неудобства. К таким предпочтениям, в частности, относятся: использование смартфона для проверки электронной почты и обмена сообщениями в мессенджерах, несмотря на неудобство ввода текста на таких устройствах; распределение элемента интерфейса прослушивания музыки на вспомогательное мобильное устройство [11]. Следовательно, при распределении элемента интерфейса необходимо учитывать такие предпочтения. Запишем это правило формально: если $a_{10}^E \neq \emptyset$, $C^D \approx \{a_{10}^E\}$ и доступно устройство D , такое, что $D \approx C^D$, то $E \rightarrow D$.

1.5. Правило конфиденциальности предусматривает, что элемент интерфейса для ввода персональных данных следует распределять на мобильное устройство, поскольку вероятность кражи введенной конфиденциальной информации на мобильном устройстве ниже, чем на персональных компьютерах и ноутбуках. Запишем это правило формально: если $a_{11}^E = 1$ и доступно устройство D , такое, что $c_3^D = 1$, то $E \rightarrow D$.

1.6. Правило ограничения распределения предусматривает, что устройство пригодно для распределения элементов

интерфейса при условии, что оно поддерживает эти элементы и соответствует им с точки зрения удобства [6]. Например, устройства Apple не поддерживают Flash-видео. Хотя разработчики браузеров нашли соответствующее решение для этих устройств, воспроизведение файлов большого объема происходит с задержками. Запишем это правило формально: если $a_{12}^E \neq \emptyset$, $C^D \approx \{a_{12}^E\}$ и доступно устройство D , такое, что $D \approx C^D$, то $E \rightarrow D$.

1.7. Правило второстепенных элементов предусматривает, что второстепенные элементы интерфейса распределяются на вспомогательные устройства. Распределение второстепенных элементов на вспомогательное устройство позволяет уменьшить высоту прокрутки страницы, что увеличит вероятность просмотра всей страницы [12]. Запишем это правило формально: если $a_8^E = 0$ и доступно устройство $D \neq D'$, такое, что $C^D \approx \{a_{11}^E\}$, то $E \rightarrow D$.

Атрибуты элементов интерфейса могут удовлетворять нескольким правилам одновременно. Тогда встает вопрос, в соответствии с каким правилом выполнять распределение. Для этого введем дополнительные правила.

1.8. Правило первостепенности логической связности предусматривает приоритет правила 1.2 над другими правилами. Как уже упоминалось, соблюдение этого правила экономит время пользователя, позволяя ему не искать зависимости между элементами интерфейса на разных устройствах.

1.9. Правило превосходства устоявшихся условностей распределения над пользовательскими предпочтениями и конфиденциальностью предусматривает приоритет правила 1.3 над правилами 1.4 и 1.5. Другими словами, если атрибуты элемента удовлетворяют правилу 1.3, то правила 1.4 и 1.5 не применяются. Ведение этого правила обусловлено приоритетом снижения временных затрат и когнитивной нагрузки на пользователя при изучении нового интерфейса.

1.10. Правило превосходства пользовательских предпочтений над конфи-

денциальностью предусматривает приоритет правила 1.4 над правилом 1.5. Другими словами, если атрибуты объекта удовлетворяют правилу 1.4, то правило 1.5 не применяется. Введение этого правила обусловлено приоритетом пожеланий

пользователя над требованиями безопасности. Впрочем, в некоторых случаях это правило может быть заменено на свою противоположность (превосходство конфиденциальности над пользовательскими предпочтениями), например в военном ПО.

2. Процедура распределения элементов интерфейса

Для распределения интерфейса в соответствии с приведенными правилами применяется процедура, состоящая из 6 шагов:

Шаг 1. Задать экспертным путем атрибуты элементов E_1, \dots, E_k интерфейса.

Шаг 2. Определить основное устройство D' в соответствии с правилом 1.1.

Шаг 3. Распределить элементы интерфейса в соответствии с правилами 1.2, 1.3 и 1.4.

Шаг 4. Распределить оставшиеся нераспределенными после шага 3 элементы интерфейса, представляющие собой левую и/или правую колонки, в соответствии с правилами 1.2, 1.5, 1.6.

Шаг 5. Распределить оставшиеся нераспределенными после шага 4 элементы интерфейса основного блока в соответствии с правилами 1.2, 1.6, 1.7.

Шаг 6. Распределить оставшиеся нераспределенными после шага 5 элементы интерфейса, представляющие собой логические блоки левой и/или правой колонок, в соответствии с правилами 1.2, 1.6.

Описанные выше правила и процедура распределения содержат значительное число компонент и логических связей между ними, поэтому сложны для анализа, внесения изменений и поиска ошибок. В связи с этим существует потребность в верификации распределения интерфейсов с целью выявления возможных ошибок.

3. Верификация распределения элементов интерфейса на основе пи-исчисления

Пи-исчисление было предложено в конце 1980-х годов Милнером [13] для моделирования взаимодействующих систем, процессы в которых имеют изменяющуюся структуру. К таким системам относятся распределенные интерфейсы.

Коммуникация процессов формально описывается с помощью аксиомы редукции:

$$x(y).P + \dots | \bar{x}\langle z \rangle.Q + \dots \rightarrow P\left\{ \frac{z}{y} \right\} Q, \quad (1)$$

где P, Q – процессы; x – канал связи; y и z – сообщения; выражение $P\left\{ \frac{z}{y} \right\}$ означает терм, получающийся из термина процесса P символьной подстановкой z вместо y .

Представим распределение интерфейса I по двум устройствам, в соответствии с процедурой, в терминах пи-исчисления на примере представленного выше интерфейса интернет-магазина (рис. 1). Процессами P будут являться: устройства – ноутбук D_1 и смартфон D_2 , на которые происходит распределение, струк-

турные блоки B_1, B_2, B_4 , логические блоки L_1, \dots, L_4 и элементы E_1, \dots, E_{18} . Процессы op ределяются через конструкцию $\stackrel{def}{=}$, слева от которой указывается имя процесса, а справа его описание.

Для управления распределением вводится процесс M , внутренним действием τ_M которого является выполнение процедуры распределения. Процесс M имеет каналы связи со всеми процессами, которые представляют устройства, а конкретно с D_1 и D_2 , как показано на рис. 3.

Шаг 1. Выполняется формирование множества элементов интерфейса $\{E_1, \dots, E_k\}$. Процесс M одновременно управляет сообщением r_elems по каналам связи md_1, md_2 с запросом информации об элементах интерфейса, которые размещены на ноутбуке D_1 и смартфоне D_2 . Процесс M имеет следующее описание в терминах пи-исчисления на данном шаге:

$$M = (\overline{md_1 \langle r_elems \rangle} | \overline{md_2 \langle r_elems \rangle}) . (\overline{md_1 \langle d_1_elems \rangle} | \overline{md_2 \langle d_2_elems \rangle}) . \tau_M . M \quad (2)$$

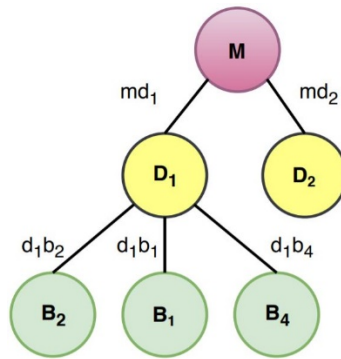


Рис. 3. Граф исходного интерфейса *I* до распределения

Процессы D_1 и D_2 получают сообщение r_elems по каналам md_1, md_2 , затем отправляют его всем процессам, с которыми они имеют связь, кроме процесса M . Процесс D_1 одновременно отправляет

$$D_1 = \overline{md_1 \langle r_elems \rangle} . (\overline{d_1b_2 \langle r_elems \rangle} | \overline{d_1b_1 \langle r_elems \rangle} | \overline{d_1b_4 \langle r_elems \rangle}) . (\overline{d_1b_2 \langle b_2_elems \rangle} | \overline{d_1b_1 \langle b_1_elems \rangle} | \overline{d_1b_4 \langle b_4_elems \rangle}) . \tau_{D_1} . \quad (3)$$

$$\overline{md_1 \langle b_2_elems + b_1_elems + b_4_elems \rangle} . D_1$$

$$D_2 = \overline{md_2 \langle r_elems \rangle} . \tau_{D_2} . \overline{md_2 \langle \{ \} \rangle} . D_2 \quad (4)$$

Процессы B_1, B_2, B_4 получают сообщение r_elems по каналам d_1b_1, d_1b_2, d_1b_4 , формируют множества элементов интерфейса $\{E_1, \dots, E_5\}, \{E_6, \dots, E_{13}\}, \{E_{14}, \dots, E_{18}\}$, где каждому элементу сопоставлены зна-

сообщение r_elems процессам B_1, B_2, B_4 по каналам d_1b_1, d_1b_2, d_1b_4 . Процессы D_1 и D_2 имеют следующее описание в терминах пи-исчисления на данном шаге:

чения атрибутов a_1^E, \dots, a_{12}^E , и отправляют их обратно процессу D_1 . Процессы B_1, B_2, B_4 имеют следующее описание в терминах пи-исчисления на данном шаге:

$$B_1 = \overline{d_1b_1 \langle r_elems \rangle} . \tau_{B_1} . d_1b_1 \langle \{E_6, \dots, E_{13}\} \rangle . B_1 \quad (5)$$

$$B_2 = \overline{d_1b_2 \langle r_elems \rangle} . \tau_{B_2} . d_1b_2 \langle \{E_1, \dots, E_5\} \rangle . B_2 \quad (6)$$

$$B_4 = \overline{d_1b_4 \langle r_elems \rangle} . \tau_{B_4} . d_1b_4 \langle \{E_{14}, \dots, E_{18}\} \rangle . B_4 \quad (7)$$

Таким образом, после выполнения всех взаимодействий процесс M формирует множество элементов исходного интерфейса I .

Шаг 2. Выполняется определение основного устройства D' . Для этого процесс M одновременно передает сообщение r_dev с запросом параметров

c_1^D, \dots, c_5^D по каналам md_1, md_2 . Далее процесс M ожидает ответы d_1_dev, d_2_dev с параметрами от процессов D_1 и D_2 . Процесс M в терминах пи-исчисления на данном шаге описывается выражением

$$M = (\overline{md_1 \langle r_dev \rangle} | \overline{md_2 \langle r_dev \rangle}) . \overline{md_1 \langle d_1_dev \rangle} | \overline{md_2 \langle d_2_dev \rangle} . \tau_M . M \quad (8)$$

Процессы D_1 и D_2 получают сообщение r_dev по каналам md_1, md_2 с за-

просом параметров c_1^D, \dots, c_5^D . В ответ они формируют сообщения, содержащие мно-

жества параметров $\{c_1^{D_1}, \dots, c_5^{D_1}\}$ и $\{c_1^{D_2}, \dots, c_5^{D_2}\}$, и отправляют их по каналам md_1, md_2 . Процессы D_1 и D_2 в терминах

$$D_1 = \overset{def}{md_1(r_dev).\tau_{D_1}.\overline{md_1}\langle\{c_1^{D_1}, \dots, c_5^{D_1}\}\rangle}.D_1 \quad (9)$$

$$D_2 = \overset{def}{md_2(r_dev).\tau_{D_2}.\overline{md_2}\langle\{c_1^{D_2}, \dots, c_5^{D_2}\}\rangle}.D_2 \quad (10)$$

После получения множеств параметров $\{c_1^{D_1}, \dots, c_5^{D_1}\}$ и $\{c_1^{D_2}, \dots, c_5^{D_2}\}$ от процессов D_1 и D_2 процесс M определяет основное устройство D' и формирует набор устройств S .

Шаг 3. Выполняется переход к следующему шагу, поскольку ни один из элементов E_1, \dots, E_{18} интерфейса I не имеет устоявшихся условностей распределения $a_9^E = \emptyset$ или пользовательских предпочтений $a_{10}^E = \emptyset$.

Шаг 4. При выполнении данного шага процесс M определил, что правая колонка B_5 отсутствует в исходном интерфейсе I , однако присутствует левая колонка B_4 . Поэтому для левой колонки B_4 выполняется поиск устройства, параметры которого соответствуют требованиям левой колонки $a_{12}^{B_4}$. Процесс D_2 объявляет новый канал связи ch , который будет использоваться для связи с элементами интерфейса, входящими в B_4 . Далее D_2 отправляет созданный канал ch процессу M по каналу md_2 и ожидает установки связи по каналу ch со структурным блоком B_4 . Процесс D_2 в терминах пи-

$$D_1 = \overset{def}{md_1(res, ch).\tau_{D_1}.\overline{((d_1b_1\langle ch \rangle + d_1b_2\langle ch \rangle + d_1b_4\langle ch \rangle) + \overline{(d_1b_1\langle ch \rangle | d_1b_2\langle ch \rangle | d_1b_4\langle ch \rangle)})}}.D_1 \quad (15)$$

В данном примере процесс D_1 отправляет сообщение res, ch процессу B_4 по каналу d_1b_4 , его описание имеет вид

$$D_1 = \overset{def}{md_1(res, ch).\tau_{D_1}.\overline{d_1b_4\langle ch \rangle}}.D_1 \quad (16)$$

Процессы B_1, B_2, B_4 ожидают от процесса D_1 по каналам d_1b_1, d_1b_2, d_1b_4 сообщений с каналом для установки новых связей. Процессы, которые получили сообщение, устанавливают связь по полученному

пи-исчисления на данном шаге описываются выражениями

исчисления на данном шаге описывается выражением

$$D_2 = \overset{def}{(vch)\overline{md_2}\langle ch \rangle.ch\langle con \rangle}.D_2 \quad (11)$$

Процесс M получает по каналу md_2 сообщение от процесса D_2 , затем отправляет сообщение res , в котором указано, для какого элемента интерфейса необходимо установить связь по новому каналу, и собственно сам канал ch . Процесс M в терминах пи-исчисления на данном шаге описывается выражением

$$M = \overset{def}{md_2(ch).\overline{md_1}\langle res, ch \rangle}.M \quad (12)$$

Применим аксиому (1) для процессов D_2 и M , тогда описание этих процессов в терминах пи-исчисления будет иметь вид

$$D_2 = \overset{def}{ch\langle con \rangle}.D_2 \quad (13)$$

$$M = \overset{def}{\overline{md_1}\langle res, ch \rangle}.M \quad (14)$$

Процесс D_1 получает сообщение res от процесса M по каналу md_1 . Процесс D_1 в терминах пи-исчисления на данном шаге описывается выражением

каналу ch . Процессы B_1, B_2, B_4 в терминах пи-исчисления на данном шаге описываются выражениями

$$B_1 = \overset{def}{d_1b_1(ch).\overline{ch}\langle con \rangle}.B_1 \quad (17)$$

$$B_2 = \overset{def}{d_1b_2(ch).\overline{ch}\langle con \rangle}.B_2 \quad (18)$$

$$B_4 = \overset{def}{d_1b_4(ch).\overline{ch}\langle con \rangle}.B_4 \quad (19)$$

В соответствии с аксиомой (1) процессы D_1 и M в терминах пи-исчисления будут иметь вид

$$M = M \quad (20)$$

$$D_1 = \tau_{D_1} . \overline{d_1 b_4} \langle ch \rangle . D_1 \quad (21)$$

В соответствии с аксиомой (1) процессы D_1 и B_4 в терминах пи-исчисления будут иметь вид

$$D_1 = D_1 \quad (22)$$

$$B_4 = \tau_{B_4} . \overline{ch} \langle con \rangle . B_4 \quad (23)$$

Таким образом, после всех взаимодействий между процессами произошло переопределение связей, что соответствует распределению левой колонки на смартфон $B_4 \rightarrow D_2$.

Шаги 5 и 6. Распределение элементов основного блока B_1 в соответствии с правилами 1.2, 1.6, 1.7 и распределение логических блоков левой B_4 и/или правой B_5 колонок в соответствии с правилами 1.2, 1.6 пропускаются, так как набор устройств S уже не содержит свободных вспомогательных устройств.

На рис. 4 представлен граф интерфейса после взаимодействий между процессами.

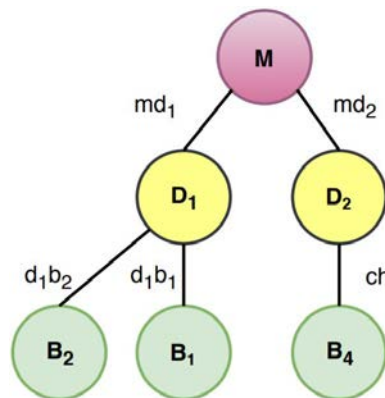


Рис. 4. Граф интерфейса после распределения

Для верификации распределения был использован инструмент PiVizTool [14]. Входными данными при этом служили описания процессов в терминах пи-исчисления (2-23). Результаты верифика-

ции показали корректность распределения: после выполнения процедуры произошло корректное переопределение связей между процессами без потери элементов интерфейса.

Заключение

В статье представлен метод верификации распределения элементов интерфейса по набору устройств, в котором для формального описания процесса распределения применен математический аппарат пи-исчисления. Метод был применен для верификации распределения по разработанной процедуре распределения элементов интерфейса по отдельным устройст-

вам, которая включает в себя пошаговый алгоритм выполнения и 10 правил распределения. Приведено пошаговое описание верификации процесса распределения интерфейса страницы интернет-магазина по разработанным правилам в терминах пи-исчисления, которые служили входными данными для инструмента PiVizTool.

Работа выполнена при поддержке гранта госзадания Минобрнауки РФ № 2.5048.2017/8.9.

СПИСОК ЛИТЕРАТУРЫ

1. Elmqvist, N. Distributed user interfaces: State of the art / N. Elmqvist // *Distributed User Interfaces*. - London: Springer, 2011. - P. 1-12.
2. Сакулин, С.А. Оптимизация интерфейсов веб-страниц с использованием инспирированных природой алгоритмов / С.А. Сакулин, А.Н. Алфимцев, Д.В. Соловьев, Д.А. Соколов // *Вестник компьютерных и информационных технологий*. - 2018. - Т. 165. - № 3. - С. 3-10.
3. Ильин, Н.И. Ситуационные центры. Опыт, состояние, тенденции развития / Н.И. Ильин, Н.Н. Демидов, Е.В. Новикова. - М.: МедиаПресс, 2011.
4. Башлыков, А.А. Принципы формирования требований к человеко-машинному интерфейсу для систем диспетчерского контроля и управления трубопроводными системами / А.А. Башлыков // *Автоматизация, телемеханизация и связь в нефтяной промышленности*. - 2014. - № 3. - С. 3-10.
5. Зацаринный, А.А. Об эргономических зависимостях между параметрами ситуационного зала с использованием изогнутого экрана / А.А. Зацаринный, К.Г. Чупраков // *Информатика и её применения*. - 2014. - Т. 8. - № 4. - С. 85.
6. Felke-Morris, T. Web Development and Design Foundations with HTML5 / Terry Felke-Morris. - 2016.
7. Yang, J. Panelrama: enabling easy specification of cross-device web applications / J. Yang, D. Wigdor // *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. - ACM, 2014. - P. 2783-2792.
8. Pribeanu, C. A Revised Set of Usability Heuristics for the Evaluation of Interactive Systems / C. Pribeanu // *Informatica Economica*. - 2017. - Т. 21. - № 3. - P. 31.
9. Krug, S. Don't make me think: Web & Mobile Usability: Das intuitive Web / S. Krug. - MITP-Verlags GmbH & Co. KG, 2018.
10. Fisher, E.R. Designing the Distributed User Interface: Case Studies on Building Distributed Applications / E.R. Fisher, S.K. Badam, N. Elmqvist // Preprint submitted to *International Journal of Human-Computer Interaction*. - 2013. - February 15.
11. Nebeling, M. Xdbrowser: User-defined cross-device web page designs / M. Nebeling, A.K. Dey // *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. - ACM, 2016. - P. 5494-5505.
12. Qiu, X. Limited individual attention and online virality of low-quality information / X. Qiu [et al.] // *Nature Human Behaviour*. - 2017. - Т. 1. - № 7. - P. 132.
13. Milner, R. Communicating and mobile systems: the pi calculus / R. Milner. - Cambridge university press, 1999.
14. Papapanagiotou, P. Formal verification approach to process modelling and composition / P. Papapanagiotou. - 2014.

14. Papapanagiotou, P. Formal verification approach to process modelling and composition / P. Papapanagiotou. - 2014.

Статья поступила в редакцию 28.06.18.

*Рецензент: д.т.н., профессор МГТУ им. Н.Э.Баумана
Деятков В.В.*

Статья принята к публикации 14.08.18.

Сведения об авторах:

Сакулин Сергей Александрович, к.т.н., Московский государственный технический университет им. Н.Э. Баумана, e-mail: ss141291@yandex.ru.

Алфимцев Александр Николаевич, д.т.н., Московский государственный технический университет им. Н.Э. Баумана, e-mail: alfim@mail.ru.

Sakulin Sergey Alexandrovich, Can. Sc. Tech, Bauman State Technical University of Moscow, e-mail: ss141291@yandex.ru.

Alfimtsev Alexander Nikolaevich, Dr. Sc. Tech, Bauman State Technical University of Moscow, e-mail: alfim@mail.ru.

Типсин Евгений Андреевич, аспирант Московского государственного технического университета им. Н.Э. Баумана, e-mail: e.tipsin@mail.ru.

Tipsin Evgeny Andreevich, Post graduate student, Bauman State Technical University of Moscow, e-mail: e.tipsin@mail.ru.