

Нестеров В.С., Безгубова Ю.К.

Аспекты использования прикладной технологии обеспечения устойчивости пользовательского интерфейса программ

Аннотация: Описываются особенности применения технологии обеспечения устойчивости структуры пользовательского интерфейса программ. Особое значение технология имеет для человеко-машинных информационно-управляющих систем, поскольку в таких системах на экране прорисовывается большое число графических элементов управления и необходимо сохранить корректную визуализацию при переносе программ в другую аппаратную среду или иную версию ОС.

Ключевые слова: пользовательский интерфейс, программа, модуль фиксации, визуальные компоненты, аппаратная среда

Для человеко-машинных информационно-управляющих систем важное значение имеет устойчивость графического интерфейса прикладных программ, поскольку на рабочей панели программ прорисовывается большое число графических элементов управления и окон отображения информации [1], а при переносе программ в другую аппаратную среду или иную версию ОС часто возникают искажения, затрудняющие работу человека-оператора. На структуру интерфейса (взаиморасположение визуальных компонентов на форме, их геометрические размеры, соотношение с используемыми шрифтами и т.п.) влияют различные параметры ОС и ее интерфейсных подсистем API и GDI+. Существенное воздействие на представление окон и текста на форме прикладной программы оказывает экранный масштаб PixelsPerInch (ppi). Из литературы [2, 3], а также из обширного массива советов (FAQ) на соответствующих сайтах в интернете [4] известны приемы, позволяющие частично снять остроту проблемы с вариабельностью графического интерфейса прикладных программ. Существуют также рекомендации [5], позволяющие улучшить ситуацию с устойчивостью графического интерфейса при переносе программного обеспечения (размещение компонентов на определенном расстоянии друг от друга, задание размеров шрифтов

в пикселях, а не в кеглях, тщательное масштабирование всех компонентов формы и т.д.). Однако в большинстве случаев применение вышеперечисленных приемов и средств не приносит желаемых результатов, а для крупномасштабных программных комплексов они просто неприемлемы. На рис. 1 демонстрирует некорректное представление визуальных компонентов формы в результате переноса программ.

На рисунке окно с текстом 1.1 является исходным для значения экранного масштаба (PixelsPerInch) $ppi=96$, а окна 1.2 - 1.5 отображают изменения, полученные при переносе программы на компьютер со значением $ppi=120$. Окно 2.1 является исходным для $ppi=120$, а окна 2.2 - 2.5 получены при переносе программы в ОС с $ppi=96$. Окна 1.4, 1.5, 2.4 и 2.5 отличаются соответственно от окон 1.2, 1.3, 2.2 и 2.3 тем, что в программе принудительно фиксируются их геометрические размеры. Для текста в окнах 1.2, 1.4, 2.2, и 2.4 задается параметр «Size» (размер шрифта в кеглях - пунктах, принятых в Windows), а размер текста в окнах 1.3, 1.5, 2.3, и 2.5 определяется параметром «Height» (размер шрифта в пикселях). Во всех случаях разрешение экрана равно 1280×1024 , используется шрифт «Times New Roman», среда разработки Embarcadero RAD Studio, ОС является Windows 10.



Рис. 1– Влияние величины экранного масштаба на визуальные компоненты формы

На рисунке окно с текстом 1.1 является исходным для значения экранного масштаба (PixelsPerInch) $ppi=96$, а окна 1.2 - 1.5

отображают изменения, полученные при переносе программы на компьютер со значением $ppi=120$. Окно 2.1 является исходным для $ppi=120$, а окна 2.2 – 2.5 получены при переносе программы в ОС с $ppi=96$. Окна 1.4, 1.5, 2.4 и 2.5 отличаются соответственно от окон 1.2, 1.3, 2.2 и 2.3 тем, что в программе принудительно фиксируются их геометрические размеры. Для текста в окнах 1.2, 1.4, 2.2, и 2.4 задается параметр «Size» (размер шрифта в кеглях - пунктах, принятых в Windows), а размер текста в окнах 1.3, 1.5, 2.3, и 2.5 определяется параметром «Height» (размер шрифта в пикселях). Во всех случаях разрешение экрана равно 1280x1024, используется шрифт «Times New Roman», среда разработки Embarcadero RAD Studio, ОС является Windows 10.

Для обеспечения устойчивости структуры интерфейса проектов и программ при смене аппаратной среды, в работе предлагается технология, которая не требует перестройки или настройки программных средств разработки под конкретный проект, осуществляется в фоновом режиме без участия программиста, обеспечивает устойчивость структуры интерфейса в большинстве случаев, в которых многие другие технологии просто не работают. В ее основе – использование файла фиксации, или файла эталонной структуры. Файл эталонной структуры создается оригинальной программой фиксации перед компиляцией программного продукта. Файл эталонной структуры не объявляется в качестве файла формы и поэтому не подвержен деформации при смене аппаратной среды или версии операционной системы. Он представляет из себя типичную подпрограмму (модуль) для среды разработки, например для Embarcadero RAD Studio, C++ Builder или Delphi. Поэтому он редактируется как стандартный файл проекта, что позволяет гибко управлять проектом – свободно настраивать интерфейс как при запуске программы, так и во время ее выполнения. При запуске программы со встроенным модулем фиксации в другой аппаратной среде или версии операционной системы, визуализация формы и компонентов на ней осуществляется согласно параметрам, сохраненным в модуле фиксации, т.е. согласно параметрам, заложенным при создании программы, независимо от настройки аппаратной среды или операционной системы. Использование параметров, заложенных в модуле фиксации, происходит после запуска программы в тот

момент, когда инструментарий операционной системы уже не воздействует на формирование типоразмеров формы и ее компонентов.

Известно, что при открытии проекта в среде с другим разрешением свойство формы PixelsPerInch будет изменено, как только будет открыта форма. Разработанная технология решает и эту проблему, используя ранее сформированный файл фиксации. При переносе проекта в другую аппаратную среду или иную версию ОС с помощью файла эталонной структуры дезавуируется автоматическая перенастройка параметров визуализации, что устраняет необходимость перестройки аппаратной среды или корректировки самого проекта. Программа фиксации выполняет необходимую корректировку файлов формы и позволяет использовать проект в имеющейся конфигурации или производить его дальнейшую разработку в новой среде. В среде Delphi создается модуль фиксации в виде pas-файла, в среде C++Builder – в виде cpp-файла с заголовочным h-файлом. В нем фиксируются параметры используемых в программе шрифтов, размеры и размещение объектов на форме и т.п. Параметры автоматически извлекаются из текстового DFM-файла формы, имеющегося в папке проекта. Для блокировки модуля достаточно дезавуировать строки, вставленные в процедуру «FormCreate». Сам модуль редактируется как стандартный файл проекта.

Описанная программа фиксации тестировалась авторами на нескольких разрабатываемых программных комплексах управления и измерения и показала несомненную целесообразность ее использования.

Литература:

1. *Гучук В.В.* Компактная визуализация динамических параметров в системах мониторинга и управления // Научная визуализация. – 2018. – № 2. – С. 61-69.
2. *Кузан Д.Я., Шаноров В.Н.* Программирование Win32 API в Delphi. – СПб.: BHV, 2013. – 368 с.
3. *Архангельский А.Я., Тагин М.А.* Программирование в C++Builder. – М.: Бином-Пресс, 2010. – 1304 с.