

Д.С. ШИШИГИН
**К ВЫБОРУ ТЕХНОЛОГИИ ИНТЕГРАЦИИ ПРИКЛАДНОГО
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С САПР**

Шишигин Д.С. К выбору технологии интеграции прикладного программного обеспечения с САПР.

Аннотация. Компьютерные программы для научных исследований со сложными геометрическими моделями целесообразно интегрировать с САПР. Рассматриваются три подхода на основе: обмена данными в формате DXF (drawing exchange format), COM-технологии и прикладной программный интерфейс (API) ObjectARX. DXF-обмен данными с САПР — простой и универсальный способ, доступный для большинства исследователей, но исключающий интерактивное управление САПР. COM технологии предоставляют простые, надежные, но не самые быстродействующие механизмы интерактивного управления САПР из внешней программы. Комбинация с DXF повышает быстродействие COM. Для задач со сложными геометрическими моделями система AutoCAD предоставляет ObjectARX API и .NET API — низкоуровневые технологии, обеспечивающие наивысшую по сравнению с другими технологиями функциональность и производительность, но имеющие ограничения. Приведены листинги программ, упрощающие понимание рассматриваемых технологий, проведен анализ их быстродействия, даны рекомендации по их применению с позиции исследователя.

Ключевые слова: геометрическая модель, САПР, AutoCAD, DXF, COM, ObjectARX, автоматизация, научная графика.

Shishigin D.S. On Choosing the Technology of Application Software Integration with a CAD-System.

Abstract. Computer programs for scientific research with complex geometric models should be integrated with a CAD-system. In the paper, we consider three approaches based on data exchange in the DXF (drawing exchange format); COM-technologies; and application programming interface (API) ObjectARX. DXF data exchange with CAD is a simple and universal way available to most researchers, but it excludes interactive CAD control. COM technologies provide simple, reliable mechanisms of interactive CAD control from an external program, although they do not have the highest level of performance. Combination with DXF increases their performance. For problems with complex geometric models, Auto-CAD-system provides ObjectARX API and .NET API – low-level technologies that ensure the maximum possible functionality and performance, compared to other technologies, but have some limitations. We present program listings that simplify the understanding of considered technologies. Their performance analysis has been conducted, and recommendations for their use are given from the researcher’s perspective.

Keywords: geometric model, CAD, AutoCAD, DXF, COM, ObjectARX, automation, scientific graphics.

1. Введение. Компьютерные программы для научных исследований, проектирования и других задач со сложными геометрическими моделями и графическими выходными данными целесообразно разрабатывать на основе интеграции с САПР. Исследователь должен писать программный код только для проведения расчетов, а ввод геометриче-

ских данных и визуализацию результатов, включая анимацию динамических процессов, можно «поручить» САПР.

По степени интеграции с САПР различают автономные программы, которые обычно только считывают данные из графических моделей, и приложения, которые работают в среде САПР. В последнем случае обычно используется AutoCAD и ведется разработка AutoCAD-приложений.

Программные способы управления AutoCAD из приложения пользователя в наиболее общем виде рассмотрены в работе [1]. Для изучения СОМ-технологий полезна работа [2]. Технологии ObjectARX посвящены работы [3, 4]. Текущую информацию по способам управления AutoCAD содержит справочная система AutoCAD.

Эти данные рассчитаны на программистов, но их явно недостаточно, чтобы сделать САПР таким же привычным инструментом для исследователей, как системы компьютерной математики (СКМ). В то же время именно интеграция прикладных программ и СКМ с САПР — один из путей автоматизации научных исследований в задачах с геометрическим моделированием.

Среди успешных примеров интеграции прикладных программ с САПР задача генерации и визуализации нестандартных 3D сеток для расчетов электромагнитных полей методом конечных разностей во временной области [5]. Расчету и проектированию линий электропередач и распределительных сетей посвящены работы [6, 7]. К сожалению, число подобных научных публикаций очень незначительно и все они посвящены только технологии ObjectARX, поэтому потенциальные возможности интеграции прикладной программы с САПР для исследователя раскрыты далеко не полностью.

При разработке программы ЗУМ [8], предназначенной для расчета молниезащиты и заземления объектов электроэнергетики, мы прошли все ступени интеграции с AutoCAD и хотели бы обобщить результаты.

Показать и сопоставить возможные технологии интеграции прикладной программы или СКМ с САПР (на примере AutoCAD) с позиции исследователя, решающего задачи со сложными геометрическими моделями, — цель настоящей работы.

2. DXF (drawing exchange format). Передача геометрических данных в программу пользователя и обратно может быть выполнена в текстовых файлах, где геометрические данные записаны в формате dxf. Структура dxf-файла документирована в справочной системе AutoCAD. Данные представлены парами «код-значение» (таблица 1), которые записываются в двух строках (без комментариев).

Таблица 1. Фрагмент dxf-файла с линией

Код	Значение	Комментарий
...	...	
0	SECTION	Начало секции
2	ENTITIES	Секция графических данных
0	LINE	Линия
8	LayerName	Название слоя с линией
...	...	
10	0	X-координата начала
20	0	Y-координата начала
30	-0.7	Z-координата начала
11	100	X-координата конца
21	100	Y-координата конца
31	-0.7	Z-координата конца
...	...	
0	ENDSEC	Конец секции
...	...	

Пусть требуется прочитать геометрическую модель, состоящую из начерченных в AutoCAD линий. Сохраним файл в AutoCAD с расширением dxf, а затем последовательно (построчно) прочитаем его в собственной программе. Точно также можно записать линии в dxf-файл (листинг 1) и построить их в AutoCAD.

```

F:=TStringList.Create; //буфер
F.Add('0'); F.Add('SECTION');
F.Add('2'); F.Add('ENTITIES');
for i:=0 to high(x1) do
begin
  F.Add('0'); F.Add('Line');
  F.Add('8'); F.Add('SetOfLines'); //зададим имя слоя
  F.Add('10'); F.Add(FloatToStr( x1[i] )); //координаты начальной точки
  F.Add('20'); F.Add(FloatToStr( y1[i] ));
  F.Add('30'); F.Add(FloatToStr( z1[i] ));
  F.Add('11'); F.Add(FloatToStr( x2[i] )); //координаты конечной точки
  F.Add('21'); F.Add(FloatToStr( y2[i] ));
  F.Add('31'); F.Add(FloatToStr( z2[i] ));
end;
F.Add('0'); F.Add('ENDSEC');
F.Add('0'); F.Add('EOF');
F.SaveToFile('F.dxf');
ADoc.ActiveDocument.Import('F.dxf',Point,1); //загрузка в AutoCAD

```

Листинг 1. Создание набора линий в формате dxf и загрузка в AutoCAD

Обмен данными через dxf-файлы имеет ряд преимуществ. Он стал стандартом обмена файлами чертежей для всех САПР, потому универсален. Обработка dxf-файлов может осуществляться с помощью любого языка программирования и с высоким быстродействием. Размерность dxf-файла может превышать миллион строк.

Этот простой способ доступа к графическим данным применяется во многих профессиональных программах, например, продукции фирмы SES (sestech.com) — мирового лидера в области расчетов заземления и молниезащиты. Он годится для любого исследователя, позволяя расширить класс решаемых в СКМ задач.

Главным недостатком обмена через dxf-файлы является отсутствие возможности интерактивного взаимодействия пользователя с чертежом. Устранить его способами более сложные способы управления САПР.

3. СОМ-технологии. AutoCAD, как и другие профессиональные САПР, разработаны с учетом концепции СОМ (СОМ — Component Object Model, модель компонентных объектов, объектная модель) фирмы Microsoft, что дает доступ к объектам AutoCAD, их свойствам, методам, событиям из других систем и их приложений и наоборот [9].

Все геометрические примитивы (линии, полилинии, точки) текущего документа (назовем его *ADoc*) находятся в коллекции *ADoc.ModelSpace*. Число примитивов дает свойство *ADoc.ModelSpace.Count*. Для чтения всех геометрических объектов чертежа из собственной программы достаточно организовать цикл (листинг 2а).

```
var StartPoint, EndPoint: OleVariant; Line: IAcadLine;
begin
  for i:=0 to Acad.ActiveDocument.ModelSpace.Count-1 do
    if Acad.ActiveDocument.ModelSpace.Item(i).ObjectName = 'AcDbLine' then
      begin
        Line:=Acad.ActiveDocument.ModelSpace.Item(i) as IAcadLine;
        StartPoint:=Line.StartPoint;
        EndPoint:=Line.EndPoint;
        ...
      end;
    end;
end;
```

Листинг 2а. Чтение координат всех линий чертежа с помощью СОМ

Метод *AddLine* позволяет нарисовать линию в AutoCAD из собственной программы (листинг 2б). Циклически применяя рассмотренный метод можно нарисовать геометрический объект, состоящих из произ-

вольного числа линий. Получаемый объект новой линии (*Line*) обеспечивает доступ к ее свойствам, что позволяет изменить цвет, слой и т.д.

```
var StartPoint, EndPoint: OleVariant; Line: IAcadLine;
begin
  StartPoint:=VarArrayCreate([0,2], varDouble);
  EndPoint:=VarArrayCreate([0,2], varDouble);
  StartPoint[0]:=0; StartPoint[1]:=0; StartPoint[2]:=0;
  EndPoint[0]:=100; EndPoint[1]:=100; EndPoint[2]:=100;
  Line:=Acad.ActiveDocument.ModelSpace.AddLine(StartPoint, EndPoint);
end;
```

Листинг 26. Создание примитива линия с помощью COM

Данный способ является полноценным средством управления AutoCAD и обеспечивает необходимый уровень интерактивности (в отличие от способа с dxf-файлами). COM поддерживает широкий спектр языков программирования, включая самые популярные: C++, Delphi, Visual Basic, Java. В большинстве сред программирования реализованы модули, упрощающие использование объектной модели, например, каркас ActiveQt для среды Qt.

Стандарт COM спроектирован для межпроцессного взаимодействия, поэтому он позволяет разрабатывать внепроцессные (out-of-process) приложения, иначе говоря, программа пользователя не является частью процесса AutoCAD, что имеет достоинства и недостатки.

Главное достоинство — это высокая универсальность. Используя позднее связывание (COM-вызовы с помощью Invoke интерфейса IDispatch), мы можем добиться того, что связывание имен свойств и методов объекта с их кодом происходит не на этапе компиляции, а на этапе выполнения программы, что позволяет использовать одно приложение для всех AutoCAD-совместимых САПР.

Однако использование механизмов межпроцессного взаимодействия и позднего связывания приводит к значительному снижению производительности (в десятки и сотни раз по сравнению с внутрипроцессными взаимодействиями и ранним связываем). Построение десятков тысяч линий может занимать минуты, что неприемлемо.

Таким образом, COM-технологии являются достаточно простым и надежным инструментом для интерактивного обмена графической информацией программы пользователя с любым САПР, пригодным для большинства научных задач. COM-технологии пригодны как для разработки приложений к САПР, так и использования в обычных программах в качестве альтернативы DXF.

Но в задачах с большими графическими данными (тысячи геометрических примитивов и более), которые требуется быстро пере-

страивать, например, при анимации динамических процессов, подход на основе COM неэффективен из-за низкой производительности.

Для разработки профессиональной программы, управляющей САПР, с полным доступом к свойствам 3D объектов рекомендуются технологии ObjectARX и .NET.

4. ObjectARX. Это система разработки программных модулей для среды AutoCAD, создающая самый низкоуровневый код по сравнению с другими средствами автоматизации AutoCAD. Код стороннего приложения выполняется в том же адресном пространстве, что и код AutoCAD, поэтому взаимодействие между ними значительно эффективнее, поскольку данные могут передаваться через указатели, а не через механизм межпроцессного взаимодействия как в COM-приложениях. Модули, разработанные в ObjectARX, работают с основными структурами данных и кодом AutoCAD и имеют доступ непосредственно к графической системе и ядру геометрических построений. В результате достигается высокая функциональность и быстродействие, но при значительно более сложной структуре программы.

Разработка пользовательских приложений (ARX-приложений) производится с использованием неуправляемого кода на языке C++, т.е. на том же языке, что использован для разработки ядра AutoCAD.

Набор разработчика ObjectARX предоставляет специальные библиотеки, а также заголовочные файлы, с помощью которых приложение пользователя получает информацию о типах, классах и объектах AutoCAD. Подключая эти файлы в свой код, программа получает возможность взаимодействовать с AutoCAD, а также создавать свои классы и объекты. Например, примитив линия описан классом *AcDbLine*. Создав объект линии, можно добавить его в пространство модели чертежа, которое представляет собой блок объектов типа *AcDbBlockTableRecord*. Данные записи, в свою очередь, расположены в таблице блоков *AcDbBlockTable*, которая, как и все остальные таблицы, принадлежит базе данных чертежа *AcDbDatabase*.

Покажем алгоритм чтения координат всех линий чертежа (листинг 3а).

Первая часть кода служит для получения блока пространства модели (*ACDB_MODEL_SPACE*), в котором хранятся примитивы чертежа. Затем с помощью итератора *AcDbBlockTableRecordIterator* происходит получение всех примитивов в цикле. При работе с объектами базы данных их требуется открывать с нужными атрибутами (*AcDb::kForRead*, *AcDb::kForWrite*, т.д.) и обязательно закрывать. В случае, если полученный объект имеет тип *AcDbLine*, он преобразовывается, и получаются его координаты.

```

AcApDocument* pActiveDocument = acDocManager->mdiActiveDocument();
AcDbDatabase* pActiveDatabase = pActiveDocument->database();
AcDbBlockTable* pBlockTable;
pActiveDatabase->getSymbolTable(pBlockTable, AcDb::kForRead);
AcDbBlockTableRecord* pBlockTableRecord;
pBlockTable->getAt(ACDB_MODEL_SPACE, pBlockTableRecord,
AcDb::kForRead);
pBlockTable->close();
AcDbBlockTableRecordIterator* pIter;
pBlockTableRecord->newIterator(pIter);
for (; !pIter->done(); pIter->step()) {
    AcDbEntity* pEnt;
    pIter->getEntity(pEnt, AcDb::kForRead);
    if (pEnt->isA() == AcDbLine::desc()) {
        AcDbLine* pLine = static_cast<AcDbLine*>(pEnt);
        AcGePoint3d startPoint = pLine->startPoint();
        AcGePoint3d endPoint = pLine->endPoint();
        ...
    }
    pEnt->close();
}
delete pIter;
pBlockTableRecord->close();

```

Листинг 3а. Чтение координат всех линий чертежа с помощью ObjectARX

Покажем способ создания линии (листинг 3б).

```

AcApDocument* pActiveDocument = acDocManager->mdiActiveDocument();
AcDbDatabase* pActiveDatabase = pActiveDocument->database();
acDocManager->lockDocument(pActiveDocument);
AcDbBlockTable* pBlockTable;
pActiveDatabase->getSymbolTable(pBlockTable, AcDb::kForRead);
AcDbBlockTableRecord* pBlockTableRecord;
pBlockTable->getAt(ACDB_MODEL_SPACE, pBlockTableRe-
cord, AcDb::kForWrite);
pBlockTable->close();
AcGePoint3d startPoint(0.0, 0.0, 0.0);
AcGePoint3d endPoint(100.0, 100.0, 100.0);
AcDbLine* pLine = new AcDbLine(startPoint, endPoint);
pBlockTableRecord->appendAcDbEntity(pLine);
pLine->close();
pBlockTableRecord->close();
acDocManager->unlockDocument(pActiveDocument);

```

Листинг 3б. Создание примитива линия с помощью ObjectARX

Сначала, как и в листинге 3а, получается блок пространства модели, в который требуется поместить примитив. Затем создается новый объект *AcDbLine* (линия) со свойствами, принятыми по умолчанию (слой, тип линии, толщина, т.д.). Эти свойства можно изменить. Линия добавляется в блок с помощью метода *appendAcDbEntity*. Перед открытием объекта для записи необходимо заблокировать рабочий документ с помощью метода *lockDocument*, а в конце разблокировать с помощью *unlockDocument*.

Циклически добавляя примитивы в пространство модели AutoCAD, можно построить геометрические модели, сетки, научную графику или чертежи любой сложности.

Высокое быстродействие операций с графическими примитивами и функциональность, обеспечивающая доступ ко всем свойствам, методам, событиям с примитивами, — главное достоинство внутри-процессных API.

Главный недостаток ObjectARX — это требование к бинарной совместимости: пользовательский модуль должен быть создан той же версией компилятора, что и AutoCAD. Компания Autodesk нарушает бинарную совместимость AutoCAD (обычно каждые 3 выпуска), поэтому требуется создавать и поддерживать несколько версий пользовательских модулей (для разных версий AutoCAD), что усложняет разработку приложений и неудобно для пользователя. Напомним, что технология COM позволяет создавать программы, пригодные для любой версии AutoCAD, и применима для любого САПР.

Технология COM допускает выбор языка программирования, что может быть важно для исследователя, в то время как ObjectARX устанавливает язык C++. ObjectARX C++ «заточен» под библиотеку MFC (Microsoft Foundation Classes), использование которой не всегда приемлемо (пакет MFC не входит в бесплатную редакцию Visual Studio Express).

Усложнение кода ObjectARX (листинг 3) по сравнению с COM (листинг 2) также относится к недостаткам с позиции исследователя. Самостоятельная разработка ARX-приложений может оказаться для него сложной задачей.

Таким образом, технология ObjectARX позволяет эффективно решать задачи в AutoCAD с геометрическими моделями и научной графикой любой сложности, но ее реализация ориентирована на квалифицированных программистов.

5. Технология .NET. На основе платформы .NET Framework — еще одно средство разработки низкоуровневых AutoCAD-приложений, имеющее некоторые преимущества перед ObjectARX. В .NET расширен выбор языка программирования (C++, C#, VB .NET и др.). Созда-

ются dll-библиотеки с управляемым кодом (управляемые сборки), что открывает программисту механизмы управления памятью, а также обеспечивает простоту разработки пользовательских интерфейсов (с помощью библиотек .NET Windows Forms или .NET Windows Presentation Foundation, входящих в состав .NET Framework).

Однако, технология .NET не в состоянии устранить главный недостаток ObjectARX — требование к бинарной совместимости: пользовательский модуль должен быть создан на той же версии целевой платформы .NET Framework, что и AutoCAD. Таким образом, переход от ObjectARX к .NET не носит принципиального характера.

6. Анализ быстродействия. Скорость построения геометрических примитивов в AutoCAD из программы пользователя — один из основных критериев, определяющих область применения рассмотренных технологий. Будем строить линии в циклах переменной размерности N с помощью программных кодов, приведенных ранее, и отмечать время выполнения (рисунок 1).

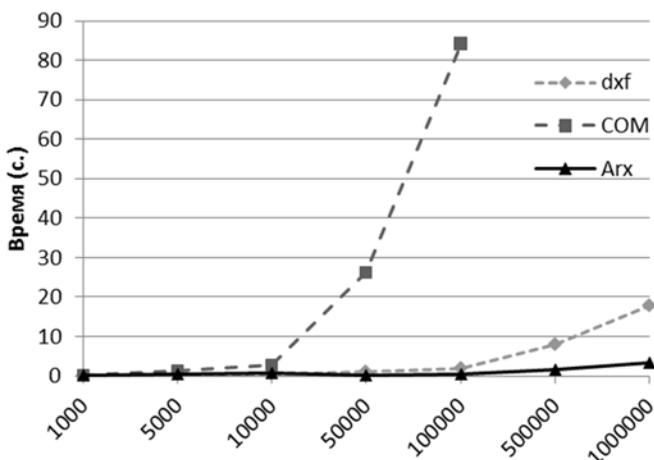


Рис. 1. Время построения линий (N – число линий) с использованием DXF, COM и ObjectARX (Компьютер: Windows 7 64-bit, Intel Core i5-4430 3.0 ГГц, ОЗУ 8 Гб)

В результате количественно подтверждены сделанные ранее выводы, а именно: межпроцессное взаимодействие через COM применимо только для задач ограниченной размерности (до 10000 геометрических примитивов). Низкоуровневые ARX-приложения обладают высокой производительностью, позволяющей проводить моделирование с большими геометрическими данными (1 млн. примитивов и более) или анимировать динамические процессы в сложных 3D системах.

Таким образом, имеем класс задач, эффективное решение которых возможно только в AutoCAD. Область применения других САПР на основе СОМ технологий сужена.

7. Комбинация СОМ с DXF. Скорость построения линий с помощью dxf сопоставима с ARX-технологией вплоть до 100000 элементов (рисунок 1), что вполне объяснимо — dxf-файл создается быстро, а его чтение — внутренние операции AutoCAD. Отсюда идея использования dxf-файла в качестве буфера обмена графической информацией с САПР на основе технологии СОМ [8]. Вместо прямого рисования линий в САПР (с использованием медленных механизмов межпроцессного взаимодействия) запишем их в dxf-файле и загрузим его в AutoCAD (листинг 1), что значительно быстрее. Этот способ позволяет использовать САПР для визуализации результатов научных исследований [8]. Он дает возможность строить графики из большего числа элементов, чем в СКМ, а послонная структура данных в САПР позволяет легко создавать выборки (включением/выключением слоев). Таким образом, использование dxf-файлов в качестве буфера обмена данными с САПР расширяет область применения СОМ-технологий.

8. Заключение. Компьютерные программы для научных исследований со сложными геометрическими моделями и задач проектирования целесообразно интегрировать с САПР. В зависимости от типа задачи возможны три основных пути: обмен данными в формате DXF, СОМ-технологии, прикладной программный интерфейс ObjectARX. DXF-обмен данными с САПР — простой и универсальный способ, доступный для большинства исследователей, но исключающий интерактивное управление САПР. Этот способ наиболее подходит для чтения геометрических данных при решении задач в СКМ. Его часто применяют в профессиональных программах с собственными геометрическими процессорами в качестве дополнительной опции.

СОМ технологии предоставляют простые, надежные механизмы интерактивного управления САПР из внешней программы, пригодные для большинства научных задач и задач проектирования. Разработка САПР-приложения на основе СОМ достаточно проста и может быть рекомендована исследователю. Низкая производительность в задачах с большими геометрическими данными — главный недостаток СОМ, однако производительность можно значительно повысить, если использовать dxf-файл в качестве буфера в операциях записи/чтения примитивов.

Для задач со сложными геометрическими моделями, анимацией динамических процессов рекомендуется система AutoCAD, которая предоставляет низкоуровневые ObjectARX API и .NET API технологии. Они обеспечивают наивысшую по сравнению с другими техноло-

гиями функциональность и производительность, хотя не лишены недостатков. Главным из них является требование бинарной совместимости программы пользователя и версии AutoCAD, другими словами, ARX-приложение работает только на определенных версиях AutoCAD, в отличие от универсальных программ, созданных по технологии COM (с использованием позднего связывания).

Литература

1. *Полещук Н.Н.* AutoCAD: разработка приложений, настройка и адаптация // СПб.: БХВ-Петербург. 2006. 992 с.
2. *Роджерсон Д.* Основы COM // М.: Изд.-торг. Дом «Русская редакция». 2000. 400 с.
3. *Kramer B.* ObjectARX Primer // Autodesk Press. 2000. 166 p.
4. *McAuley C.* Programming AutoCAD 2000 Using ObjectARX // Autodesk Press. 2000. 678 p.
5. *Jun Y., Dong-Lin S., Xiao-Ying Z.* The study and realization of automatic mesh generation based on electromagnetic simulation of FDTD // 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications. 2009. pp. 1242–1245.
6. *Hou X., Lu X.* Automatic design of transmission lines using ObjectARX technology. International Conference on Control Engineering and Communication Technology (ICCECT). 2012. pp. 581–585.
7. *Huang Y. et al.* Design and implementation of reliability evaluation module for medium voltage distribution networks by objectarx technology // 8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009). 2009. pp. 1–6.
8. *Шнишгин Д.С.* AutoCAD приложение для расчета молниезащиты и заземления объектов электроэнергетики // Автоматизация в промышленности. 2014. № 9. pp. 28–32.
9. *Полещук Н.Н.*, Программирование для AutoCAD 2013–2015 // М.: ДМК Пресс. 2015. 462 с.

References

1. *Poleshuk N.N.* AutoCAD: *razrabotka prilozhenij, nastrojka i adaptacija* [AutoCAD: application development, configuration and adaptation]. SPb.: BHV-Peterburg. 2006. 992 p. (In Russ.).
2. *Rodzherson D.* *Osnovy COM* [COM basics]. M.: Izd.-torg. Dom «Russkaja redakcija». 2000. 400 p. (In Russ.).
3. *Kramer B.* ObjectARX Primer. Autodesk Press. 2000. 166 p.
4. *McAuley C.* Programming AutoCAD 2000 Using ObjectARX. Autodesk Press. 2000. 678 p.
5. *Jun Y., Dong-Lin S., Xiao-Ying Z.* The study and realization of automatic mesh generation based on electromagnetic simulation of FDTD. 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications. 2009. pp. 1242–1245.
6. *Hou X., Lu X.* Automatic design of transmission lines using ObjectARX technology. International Conference on Control Engineering and Communication Technology (ICCECT). 2012. pp. 581–585.
7. *Huang Y. et al.* Design and implementation of reliability evaluation module for medium voltage distribution networks by objectarx technology // 8th International Confer-

ence on Advances in Power System Control, Operation and Management (APSCOM 2009). 2009. pp. 1–6.

8. Shishigin D.S. [AUTOCAD application for calculation of lightning protection and grounding of electric power industry objects]. *Avtomatizacija v promyshlennosti – Automation in industry*. 2014. vol. 9. pp. 28–32. (In Russ.).
9. Poleshchuk N.N. *Programmirovanije dlja AutoCAD 2013–2015* [Programming for AutoCAD 2013-2015]. M.: DMK Press, 2015. 462 p. (In Russ.).

Шишигин Дмитрий Сергеевич — аспирант кафедры автоматизации и вычислительной техники Вологодского государственного университета. Область научных интересов: разработка программного обеспечения в области молниезащиты и заземления объектов электроэнергетики. Число научных публикаций — 25. shishigind@yandex.ru; ВоГУ, ул. Ленина, д. 15, г. Вологда, 160000, РФ; т. +79210646334.

Shishigin Dmitry Sergeevich — postgraduate student of the Automation and Computer Engineering department of the Vologda State University. Research interests: software development in the area of lightning protection and grounding of electric power industry objects. The number of publications — 25. shishigind@yandex.ru; VSU, 15, Lenina str., Vologda, 160000, Russia; p.n. +79210646334.

РЕФЕРАТ

Шиншигин Д.С. **К выбору технологии интеграции прикладного программного обеспечения с САПР.**

Компьютерные программы для научных исследований со сложными геометрическими моделями и задач проектирования целесообразно интегрировать с САПР. В зависимости от типа задачи возможны три основных пути: обмен данными в формате DXF, СОМ-технологии, прикладной программный интерфейс (API) ObjectARX. DXF-обмен данными с САПР — простой и универсальный способ, доступный для большинства исследователей, но исключающий интерактивное управление САПР. Этот способ наиболее подходит для чтения геометрических данных при решении задач в СКМ. Его часто применяют в профессиональных программах с собственными геометрическими процессорами в качестве дополнительной опции.

СОМ технологии предоставляют простые, надежные механизмы интерактивного управления САПР из внешней программы, пригодные для большинства научных задач и задач проектирования. Разработка САПР-приложения на основе СОМ достаточно проста и может быть рекомендована исследователю. Низкая производительность в задачах с большими геометрическими данными — главный недостаток СОМ, однако производительность можно значительно повысить, если использовать dxf-файл в качестве буфера в операциях записи/чтения примитивов.

Для задач со сложными геометрическими моделями, анимацией динамических процессов рекомендуется система AutoCAD, которая предоставляет низкоуровневые ObjectARX API и .NET API технологии. Они обеспечивают наивысшую по сравнению с другими технологиями функциональность и производительность, хотя не лишены недостатков. Главным из них является требование бинарной совместимости программы пользователя и версии AutoCAD, другими словами, ARX-приложение работает только на определенных версиях AutoCAD, в отличие от универсальных программ, созданных по технологии СОМ (с использованием позднего связывания).

SUMMARY

Shishigin D.S. **On Choosing the Technology of Application Software Integration with a CAD-System.**

Computer programs for scientific research with complex geometric models and for design problems should be integrated with CAD-systems. Depending on the problem, three main ways are possible: data exchange in the DXF format, COM-technologies, application programming interface (API) ObjectARX. DXF data exchange with CAD is a simple and universal way available to most researchers, but it excludes interactive CAD control. This approach is more appropriate for geometric data reading when solving problems in computer mathematics systems. It is often used in professional programs with their own geometric processors as an extra option.

COM technologies provide simple, reliable mechanisms of interactive CAD control from an external program, suitable for most research and design problems. CAD-application development based on COM is simple enough and can be recommended to the researcher. Low performance in problems with large geometric data is the main drawback of COM, however, the performance can be significantly increased if you use DXF file as a read/write buffer for geometric entities.

For problems with complex geometric models, dynamic processes animation AutoCAD is recommended which provides low-level ObjectARX API and .NET API technologies. They ensure the maximum possible functionality and performance, compared to other technologies, but have some drawbacks. The main among them is the binary compatibility requirement of user application and AutoCAD version. In other words, ARX-application works only with the certain AutoCAD versions, unlike universal programs based on the COM technology (late binding).