



Обзор стандарта IEC 61499

*Гулько С.В., "ХОЛИТ Дэйта Системс", г.Киев
Николаас Джоврей, ICP Triplex ISaGRAF, Франция*

В среде профессиональных инженеров существует своего рода традиция - обсуждать, порой очень горячо, различные подходы и парадигмы в проектировании, программировании, и еще во многих разных сферах. Зачастую эти споры ведутся не с целью убедить оппонента перейти в другой лагерь и воспользоваться новой технологией, а преследуют цель потешить собственное самолюбие. В результате спора стороны расходятся в состоянии глубокого удовлетворения собственными знаниями, но иногда, а это бывает крайне редко, все же зарождается сомнение - а настолько я прав в своей позиции и убеждениях? Авторы не раз бывали участниками подобных диспутов, один из которых показал преимущество объектных технологий. Пойдите, при чем здесь объектные технологии, спросите Вы? Все очень просто - с появлением стандарта IEC 61499 они войдут в сферу автоматизации производства.

Идея воспользоваться объектным отображением процесса далеко не нова - в настоящий момент она успешно используется при разработке больших программных комплексов, что подтверждает ее реальную применимость на практике.

Объекты можно использовать для описания любых физических явлений и процессов, они представляют собой гибкий описательный инструмент. Сочетание гибкости и простоты дает возможность описывать в терминах объектов сложные явления, упростить которые можно путем выделения и обобщения их базовых свойств. Все это может показаться немного странным и надуманным, однако авторы придерживаются именно этой точки зрения, поэтому ниже этот вопрос будет рассмотрен более детально.

Упрощение труда связано с совершенствованием инструментов и

методологий. За примерами далеко ходить не надо: сейчас можно быстро создать распределенное приложение, выполняющееся на нескольких процессорных мощностях с синхронизацией по сети Ethernet. Еще вчера на создание такой системы можно было потратить годы - сегодня же это доступно всем. Правда, такое упрощение для одних оборачивается огромным трудом для других. Но это, как говорится, уже другая история.

Распределенные системы были упомянуты отнюдь не случайно. Они в наибольшей степени отражают совершенствование современных технологий, в том числе и в автоматизации производственных процессов, характеризуют переход от сосредоточенных систем, исполняющихся на одном вычислителе, в котором реализуются алгоритмы и программы, к мобильным и компактным компонентам, функционирующим в разветвленной сети. Распределенные системы начали пробивать себе дорогу уже давно, по мере чего выявлялись проблемы и предпринимались попытки их разрешения. Наиболее острой проблемой, безусловно, стала синхронизация и обмен данными между узлами. На рынке существует большое число решений, позволяющих в той или иной степени решить эти проблемы. Это и CORBA, и DCOM, и всевозможные RPC, RMI и... перечислять можно долго. Но много - это не значит хорошо. Функции и внутренние данные одного приложения зачастую остаются недоступными для другого приложения, написанного на том же языке программирования и запущенного на том же компьютере. Огромное количество времени и ресурсов тратится на построение интерфейсов между всевозможными устройствами, написание драйверов, согласование сигналов. Порой создается впечатление, что поставщики аппаратного и программного обеспечения живут в своеоб-

разном вакууме, не зная (или не желая знать), что кроме них в мире существует кто-то еще! В менеджменте существует такое правило - хороший управляющий строит работу на сильных качествах людей, на том, что они умеют делать лучше всего. Таким же принципом можно было бы руководствоваться и в сфере автоматизации производства - формировать систему из узкоспециализированных компонентов, строить ее по кирпичикам, тщательно отбирая наиболее удачные и подходящие решения. Именно эта идея использования наиболее совершенных функциональных блоков и лежит в основе стандарта IEC-61499.

История возникновения IEC-61499

Стандарт имеет долгую историю, которая началась в 90-х годах, когда Международная Электротехническая Комиссия (IEC) получила Рабочее предложение стандартизировать некоторые аспекты применения программных модулей, называемых "функциональные блоки", в промышленно-ориентированной распределенной измерительной и управляющей среде. Будущий документ был условно разделен на четыре части, некоторые из которых детальнее рассмотрим ниже. В проект были положены такие стандарты, как IEC-61131 и IEC-61158 (Fieldbus). После перипетий, присущих скорее приключенческому роману, чем столь уважаемой и серьезной организации как IEC, по прошествии почти 10 лет появилась первая часть стандарта, а через год - вторая. Этого уже было достаточно, чтобы начинать работу по внедрению нового стандарта в реальных продуктах.

Функциональные блоки призваны инкапсулировать в себе реализацию тех или иных функций и представлять интерфейсы для взаимодействия с другими блоками. Таким об-



разом происходит переход на более высокий уровень абстракции, где оперируют понятиями объектов производства. Если перейти от маловразумительных теоретических выкладок к человеческому языку, то мы получим следующее - интеллектуальные устройства (например, турбина, насос, датчик, контроллер, панель пользовательского интерфейса) будут обеспечивать часть функциональности, из которой будет создаваться общая схема взаимодействия. Например, регулятор-ползунок, как элемент интерфейса пользователя на панели управления, может быть напрямую подключен к насосу. А ведь можно скрыть реализацию глубже, перейдя от уровня насосов, предположим, к цехам или, еще лучше, к заводам!

Естественно, чтобы обеспечить такие возможности коммуникаций, должна быть проделана огромная работа - как со стороны разработчиков программного обеспечения, так и со стороны поставщиков оборудования. Найти общий язык, наладить интерфейсы, не потеряться в спорах, чей метод более правильный, призван стандарт IEC 61499.

Методология создания проекта на основе IEC-61499

Стандарт IEC-61499 определяет общую модель и методологию, применимую для описания функциональных блоков в формате, независимом от реализации. Эта методология может быть использована разработчиком для создания распределенной

функциональности на физические устройства, например, на промышленные контроллеры или модули ввода-вывода.

Если подходить к процессу построения системы "по науке", то первая стадия предполагает применение так называемых P&ID (Process and Instrumentation Diagrams) - структурных схем, отражающих в некотором роде суперпозицию физического процесса или объекта и применяемых в нем технических средств. На создаваемой схеме наносятся различные технологические установки, указывается расположение датчиков и т.д. Этот документ является своеобразной отправной точкой, началом проекта, и индикатором уровня профессионализма, причем не только разработчика системы, но и заказчика проекта. От того, насколько грамотно выполнена эта стадия проекта, во многом определяется его конечный успех и, как говорится, это есть условие "необ-



управляющей системы. Система, описанная в рамках логически соединенных блоков, может быть выполнена на различных вычислительных ресурсах.

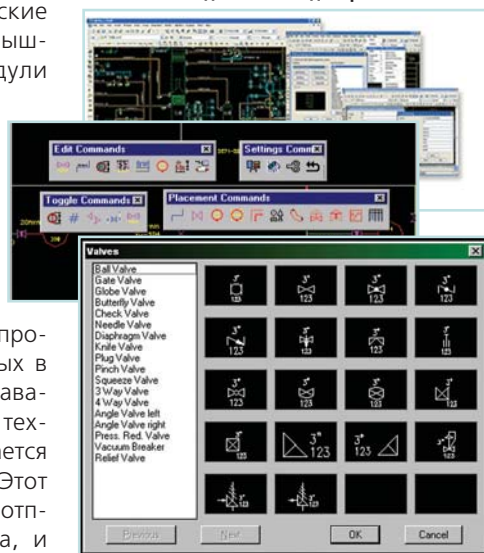
Условно можно выделить три стадии создания проекта в терминах функциональных блоков - первичный анализ на физическом уровне, выделение областей функциональной зависимости и их взаимодействия, и отображение

необходимое, но недостаточное". В настоящий момент существует огромное количество программ разного уровня и ориентации, облегчающих создание P&ID - среди них можно упомянуть AutoCAD, CADWorx, SmartPlant и многие другие.

После того, как была создана P&ID схема, можно приступать к построению зависимостей между различными элементами. Этот шаг подразумевает связывание потоков данных, проходящих между всеми логическими единицами, определенными на предыдущем этапе.

Наконец, этап отображения представляет собой процесс перевода созданной архитектуры в термины функциональных блоков, а затем - перенос на аппаратные средства, подстройку средств ввода/вывода, согласование коммуникационных каналов и некоторые другие действия.

Создание P&ID диаграмм



Структура стандарта IEC- 61499

Стандарт разделен на четыре части, доработка которых велась достаточно долго. Объяснить это можно глобальностью самой идеи, ведь стандарт рассчитан не только на производителей различных контроллеров и разработчиков программного обеспечения. Вся притягательность, все положительные стороны этого стандарта проявятся только в случае, если будут выпускаться стандартные (с точки зрения производства) устройства, такие как котлы, печи, станки, насосы, обладающие определенным набором качеств, обеспечивающих совместимость с требованиями IEC-61499.

Производители промышленных систем управления, поставщики обо-

рудования из США, Японии, Англии, Европы вошли в рабочую группу IEC, и к концу 2000-го года появилась первая часть стандарта, описывающая архитектуру функциональных блоков, подходы к созданию и применению их на практике. Этот документ содержит описание следующих ключевых позиций:

- правила описания функциональных блоков;
- правила поведения событий в функциональных блоках;
- правила использования функциональных блоков в конфигурации распределенной промышленной системы;
- правила взаимодействия функциональных блоков по различным каналам связи;
- правила использования функциональных блоков при управлении приложениями, ресурсами и устройствами в распределенных управляющих системах;
- требования соответствия стандартам.

Как можно заметить, первая часть носит скорее теоретический характер, лишь описывая, что же есть "функциональные блоки", ни словом не упоминая о деталях реализации. А ведь именно реализация, как в коде, так и в "железе", будет играть решающую роль в будущем стандарта - найдет ли он отклик у поставщиков оборудования и программных продуктов как решение продуманное и удобное, или станет всего лишь необходимым для соблюдения принципа "у них есть, а чем мы хуже".

Вторая часть стандарта IEC-61499, опубликованная летом 2001 года, дает ответы на эти животрепещущие вопросы. Авторы пошли по наиболее простому пути, решив не изобретать еще раз то, что уже существует. Чтобы описать, какие именно внешние интерфейсы обеспечивает функциональный блок, какие события он генерирует, а какие ожидает сам, чтобы потом на них отреагировать заранее заданным способом, задать входные и выходные сигналы, определить графическое отображение - для всего этого используется XML. Extensible

Markup Language (Расширяемый язык разметки) - средство, позволяющее структурно описать любой тип данных в текстовом виде для дальнейшего хранения и транспортировки. Основное преимущество - кроссплатформенность. Данные, закодированные в XML, будут одинаково интерпретированы везде.

Однако отображение - это лишь вершина айсберга. В глубине, за простым изображением может скрываться огромный объем кода, делающего сложные вещи простыми. Для того, чтобы "функциональный блок" действительно стал функциональным, разработчик должен проделать работу по созданию алгоритмов или функций, которые придадут его объекту необходимые свойства. Для того, чтобы задать эти свойства, можно будет воспользоваться языками стандарта IEC-61131 (SFC, FB, LD, ST, IL), что, безусловно, очень удобно. Новый стандарт в основе своей будет иметь именно эти, проверенные временем, технологии программирования. Таким образом обеспечивается преемственность решений, когда тысячи разработчиков смогут легко перейти к созданию программ по новой методологии - объектно-ориентированной.

Кроме стандартных языков IEC 61499, новые функциональные блоки можно будет создавать и на стандартных языках программирования, таких как C++ или Delphi.

Части 3 и 4 до сих пор находятся в стадии доработки и согласования. Приблизительная дата их опубликования - 2007 и 2010 года соответственно. В них должны быть детально рассмотрены вопросы сетевого взаимодействия между компонентами, требования соответствия стандартам и т.д.

Весь мир - объекты

Предложенная давным-давно парадигма объектного программирования доказала свою жизнеспособность, обретя огромное число приверженцев, получив реализацию в десятках языков, в сотнях тысяч библиотек. Случилось это потому, что

создатели идеологии изначально исходили из предпосылок, что человеку гораздо легче оперировать понятными ему вещами - объектами реального мира. Согласитесь, удобнее работать с объектом Стул, Стол, Шкаф и Комната, хранящими внутри себя все основные свойства - положение в пространстве, цвет, материал и т.д., чем с набором разрозненных абстрактных переменных!

Кроме того, что объекты отражают элементы реального мира, они обладают таким, безусловно, полезным свойством, как стабильность. Окружающий нас мир стабилен, и в тоже время пластичен. Это свойство проявляется в том, что сам по себе, без внешнего воздействия, он меняться не будет. Подобное высказывание справедливо и для объектов (и для функциональных блоков) - они есть устойчивая сущность, изменяющая свое состояние лишь в результате заранее предопределенного воздействия.

Наиболее ценное, что есть у работника интеллектуального труда - его опыт. Полученные ранее знания он применяет в своей работе, опирается на них, когда нужно принять решение. В сегодняшнем динамично меняющемся мире невозможным ресурсом является только время. Профессионал ценит свое личное время и умеет рационально его использовать. Так уж получилось, что большая часть работы, которую приходится выполнять разработчику - рутинная. К счастью, рутину можно упаковать в объект, чтобы потом воспользоваться им, экономя время. Удачное решение, которое потенциально можно применить еще раз в другой задаче, еще больше заслуживает вынесения в отдельный объект. В последнем случае требуется умение видеть и выделить общие, присущие не только данному проекту, детали. Эта эссенция сущности действительно способна творить чудеса с рабочим процессом - построение приложения на основе готовых компонент, объектов, часть из которых создана самим разработчиком - увлекательнейшее занятие, своеобразное искусство. IEC-61499 даст Вам возможность творить из определенных компонентов, видеть архитектуру проекта в целом.

Далеко не всем хочется испытывать своего рода творческий экстаз, люди предпочитают просто работать. Отлично, объекты позволяют не вникать в детали, в тонкости реализации того или иного алгоритма. Поставщик

```

1 <project name="test">
2   <device id="1" type="PAC">
3     <vendor name="IPCDAS" />
4     <model name="i-8837" />
5     <communication type="ethernet">
6       <address port="192.168.0.1" port="1234" />
7     </communication>
8   </device>
9   ...

```

XML - выражение

интеллектуального оборудования даст Вам готовые библиотеки и документацию, описывающую интерфейсы его функциональных блоков. Интегратор быстро собирает на их основе приложение, сдает объект и едет заниматься своим любимым делом, например, кататься на лыжах. Немножко упрощенно, но в целом верно. Новые функциональные блоки позволят не тратить время на детальное изучение объектов, важно знать лишь интерфейс.

Пожалуй, единственное, что не было перенято у объектно-ориентированного программирования - это механизмы наследования. Наследование позволяет передавать некоторые свойства от родителя потомку. Например, для объекта Мебель свойством будут материал. Объект Кресло будет порожден от Мебель, может иметь массу своих специфичных свойств (спинка, подлокотники и т.д.), но также будет иметь и часть родительских свойств. Почему он не вошел в стандарт - остается только гадать, однако не следует забывать, что две части IEC-61499 все еще находятся в доработке.

Анатомия функциональных блоков

Для того чтобы определить функциональный блок, нужно указать его следующие свойства: внешний интерфейс (входные и выходные переменные), схему управления и вызываемые ею алгоритмы. Схема управления, по сути, представляет со-



бой карту реакций на внешние воздействия, в которой расписывается, что именно нужно сделать, если произошло конкретное событие. Помимо внешних воздействий в карте указывается (при необходимости), какие события будут генерироваться при окончании выполнения функций-работчиков. Наиболее удобным язы-

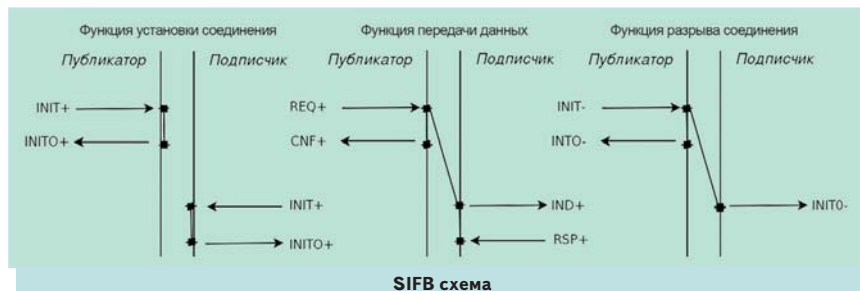
ком для создания схем управления является SFC, в котором можно достаточно легко сделать привязку вызова определенных функций в ответ на изменения некоторых переменных. Алгоритмы обработки событий можно создавать на любом из языков IEC-61131 или на любом другом, способном воспринимать и обрабатывать входные, выходные и внутренние переменные. Можно не сомневаться, что появится большое количество библиотек, позволяющих задавать алгоритмы функциональных блоков на стандартных языках программирования типа C или Delphi.

Функциональные блоки также поддерживают знакомое многим из объектно-ориентированного проектирования понятие интерфейса обслуживания. Этот механизм позволяет указать, какие именно последовательности действий должны быть проведены, чтобы получить адекватный результат. С его помощью можно

■ в программировании уже давно укоренилось понятие проектирования и в этой области существует огромное количество разработок и методологий.

Вывод из этого можно сделать следующий - можно ожидать появления возможностей создания проектов на основе функциональных блоков также и в программных пакетах типа Rational Rose или других CASE-инструментах.

Если развить мысль дальше, например, в направлении автоматизации и управления на основе web-технологий, то и тут связь функциональных блоков и XML обеспечивают выгодную позицию, так как уже сейчас есть браузеры, способные корректно отображать поступающую из сети информацию в этом формате. Идея построения легковесного пользовательского интерфейса, независимого ни от платформы, ни от расстояния, централизованного и легкоуправляемого, витает в воздухе уже давно. Что



легко описать как операции ввода/вывода для различных технических средств, так и взаимодействия со всевозможными службами запуска по времени или таймерами.

Согласно второй части стандарта IEC-61499 все описания функциональных блоков сохраняется в формате XML. Чтобы обеспечить стандартизацию библиотек, были разработаны соответствующие DTDs (Document Type Definitions - определители типа документа). Механизмы DTD являются единственным стандартным способом описания грамматики документа в формате XML, в нем фиксируются допустимые для применения типы данных, физические устройства, дополнительные ресурсы и файлы конфигурации.

Можно выстроить следующую цепочку:

■ в настоящий момент XML является стандартом de-facto для обмена информацией между приложениями;

■ функциональные блоки являются аналогами классов в объектно-ориентированном программировании;

даст IEC-61499 для развития этого направления пока говорить рано, но кое-какие выводы сделать можно уже сейчас - это ориентация на распределенные системы и, в чем-то, на коммуникационные протоколы типа Ethernet.

Реакция на события

Как уже упоминалось, реальный мир пластичен, он меняет свое состояние лишь при внешнем воздействии. Такой же принцип положен и в работу функциональных блоков - они стали событийно-зависимыми. В чем же превосходство событийной модели выполнения? Какие преимущества она способна дать по сравнению с другими системами (да и какие они есть, эти другие)?

В IEC-61131 программы выполнялись в детерминированном порядке - одна команда следовала за другой. Также можно было построить приложение, выполняющееся с какой-то периодичностью в конечном или замкнутом цикле. По большому счету, ничего очень уж неудобного в такой сис-

теме нет, ведь подобная схема является следствием внутренней организации тех операционных систем, где выполняется задача. Просто более естественным решением было бы взаимодействие, где входящее воздействие порождает некую реакцию со стороны объекта, к которому оно прикладывается. Таков порядок вещей в окружающем нас мире, а практика показывает, что наиболее жизнеспособные решения переносят концепции реального мира в технику.

В терминах событийной модели можно описать любую стратегию выполнения программы и неважно, будет ли это циклическая структура или схема, основанная на синхронизации по времени - все они отлично укладываются в общую модель.

Функциональный блок, созданный по стандарту IEC-61499, сможет как принимать, так и порождать события. Например, соединенные операторская панель HMI (human-machine interface) и интеллектуальный насос будут производить свое общение на основе событий: перемещение оператором регулятора-ползунка на экране монитора HMI сгенерирует событие, реакцией на которое будет, предположим, увеличение или уменьшение оборотов двигателя.

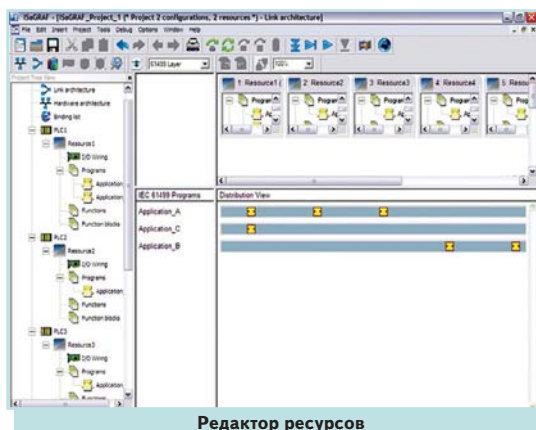
Стремясь оперировать только общими понятиями, создатели стандарта ограничились лишь одним типом событий, не проводя никаких разделений по каким-либо признакам, возложив классификацию на плечи разработчиков. В своей программе, инкапсулирующей логику блока, разработчик будет проводить отбор необходимых ему событий, дабы обрабатывать только значимые из них.

ICS Triplex ISaGRAF 5.0

Стандарт IEC-61499 уже давно находится в разработке, и до ее окончательного завершения осталось пару лет. Такие сроки вполне объяснимы, если вспомнить глобальность поставленной задачи, а в глобальных задачах, как известно, нет права на ошибку. Однако уже сейчас две первых части стандарта получили статус открытой для общества спецификации PAS (Publicly Available Specifications), что позволяет закладывать поддержку грядущих нововведений в свои программные и аппаратные продукты.

Программный продукт фирмы ISC Triplex ISaGRAF является бесспорным лидером в области программных систем класса SoftLogic, а лидер, как

известно, должен нести вперед знания прогресса. Итак, 15-го декабря 2005 года появился первый в мире програ-



Редактор ресурсов

ммный продукт, позволяющий создавать проекты, отвечающие требованиям стандарта IEC-61499.

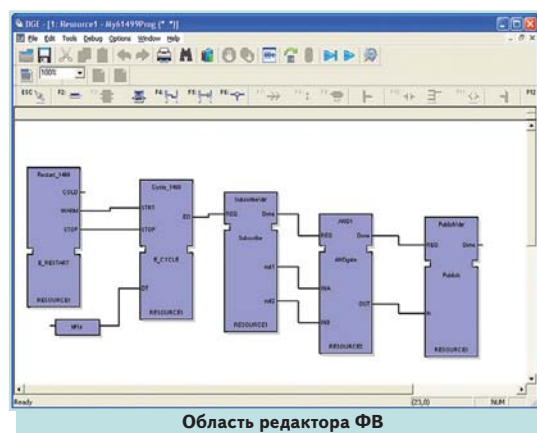
ISaGRAF версии 5.0 объединяет в себе уже известные возможности IEC-61131 и нововведения, присутствующие функциональным блокам, о которых уже говорилось выше. Эти возможности позволяют создавать приложения, объединяющие многие ресурсы, распределенные на различные исполнительные устройства, создавать сложную вычислительную систему из уже готовых функциональных блоков или, создавая свои собственные блоки, закладывая в них именно ту функциональность, которую они призваны обеспечить.

В сравнении с предыдущими версиями, программа претерпела серьезные изменения, причем как внешние, так и внутренние. Следует особо отметить, что сохранилась совместимость с исполняемыми системами 4-ой версии, что позволит с ее помощью создавать проекты и для более ранних версий ISaGRAF 4.xx.

Взгляд в будущее

Сейчас еще рано говорить о состоявшемся приходе на рынок нового стандарта, можно лишь сказать, что первые кирпичики уже заложены. Развитие IEC-61499 будет в равной степени зависеть как от разработчиков средств автоматизации, так и от создателей различных технологических (если рассматривать только эту отрасль) установок, и также от конечных потребителей - системных интег-

раторов, инженеров, программистов, которым удобно иметь дело с интеллектуальной техникой. Заинтересованных лиц много, это верно, как верно и то, что объем работы, которую предстоит проделать, чтобы перенести все возможности нового стандарта с бумаги в реальность, просто огромен. Это не дело ближайших нескольких месяцев или даже лет, но эта работа будет сделана. Этого требуют хотя бы соображения эволюции и обновления технологий, не говоря уже о здравом смысле.



Область редактора ФВ

У нашей страны огромный потенциал, как интеллектуальный, так и технологический. Постепенно приходит понимание того, что следует делать акцент на собственное производство, а чтобы конкурировать с игроками мирового уровня, необходимо делать это на должном уровне. Пройдет еще какой-то период времени и под "должным уровнем" будут понимать поддержку стандарта IEC-61499. Сейчас у нас есть возможность оказаться в начале процесса, обеспечив себе выгодное положение пионеров. Или же можно подождать, но потом снова и "как всегда" оказаться в роли договоряющихся. Выбор за Вами!



КОНТАКТЫ:

т. (044) 493-31-08, 492-31-09
e-mail: info@isagraf.com.ua