

Программная реализация генерации чертежей AutoCAD в веб-приложении

В. В. Осокин, В. А. Бендик, З. А. Ниязова

В статье рассматриваются задачи генерации спецификаций, характеристик и схем вентиляционных установок в формате dxf, а также генерации собранных установок в 3D-формате. Данный функционал должен быть реализован на веб-технологиях, поскольку будет встраиваться в веб-приложение по расчёту вентиляционного оборудования. В частности, должна быть реализована возможность вращения 3D-модели установки в браузере.

Ключевые слова: генерация чертежей, AutoCAD, openjs-cad, 3D, dxf, вентиляционные установки.

Введение

В сфере производства вентиляционного оборудования важной задачей является расчёт вентиляционных установок. Результат расчёта оборудования используется не только менеджерами по продажам, которых больше интересует расчётная цена этого оборудования, но также проектировщиками и сборщиками оборудования.

Проектировщикам при проектировании здания требуется заложить в проект вентиляционное оборудование. Проект обычно выполняется в программе AutoCAD. В связи с этим актуальной задачей является генерация спецификации, характеристик и схем вентиляционных установок в формате dxf.

Кроме того, сборщикам вентиляционных установок требуется видеть, как будет выглядеть уже собранная установка. В связи с этим, ещё одной актуальной задачей является генерация собранной вентиляционной установки в 3D-формате. Важным требованием является возможность быстрого просмотра сгенерированного файла в веб-браузере.

Постановка задачи и полученные результаты

Ставится следующая задача. По исходным данным отрисовать в векторном формате чертежи (см. рис. 2), таблицу спецификации (см. рис. 1), таблицу характеристик (см. рис. 9) в формате dxf, а также трёхмерное изображение установки в 3d-формате (см. рис. 13).

Позиция	Наименование и техническая характеристика	Тип, марка обозначение документа, опросного листа	Код оборудования, изготовитель, материала	Завод-	Единица Изменения	Количество
1	2	3	4	5	6	7
	<i>Вентиляция</i>					
1	<i>Приточно-вытяжная установка в составе</i>					
1)	<i>Гибкая вставка</i>	<i>Б.1 (2)</i>			<i>шт.</i>	<i>1</i>
2)	<i>Клапан воздушный</i>	<i>КВ.1 (2)</i>			<i>шт.</i>	<i>1</i>
3)	<i>Карманный фильтр КФ.4</i>	<i>Ф025 КФ.4 (Н/Д)</i>			<i>шт.</i>	<i>1</i>
4)	<i>Теплоутилизаторы пластинчатый</i>	<i>ТУП (Ш1000/5.0/Е)</i>			<i>шт.</i>	<i>1</i>
5)	<i>Нагреватель водяной</i>	<i>HEATING 6.1</i>	<i>5.25.HB.Y...</i>		<i>шт.</i>	<i>1</i>
6)	<i>Вентилятор</i>	<i>ЭР50Ф-4.ДНД1 Р</i>			<i>шт.</i>	<i>1</i>
7)	<i>Гибкая вставка</i>	<i>ГВ.2 (2)</i>			<i>шт.</i>	<i>1</i>
8)	<i>Гибкая вставка</i>	<i>ГВ.2 (2)</i>			<i>шт.</i>	<i>1</i>

Рис. 1. Спецификация вентиляционной установки.

Для реализации программной генерации чертежей и таблиц используется библиотека DXFwriter, исходный код можно найти по адресу <https://github.com/digitalfotografen/DXFwriter>.

Для реализации генерации трёхмерных установок (см. рис. 13) используется библиотека OpenJsCad, исходный код которой можно найти по адресу <https://github.com/Spiritdude/OpenJSCAD.org>.

Реализован модуль отрисовки, который может генерировать САД-файлы в формате DXF. Он может формировать разные типы векторной графики: линия, текст, окружность. Получая на входе json строку с данными, формируется dxf-файл, который после генерации передается для скачивания. Также реализован модуль генерации трёхмерной установки для отображение в веб-браузере.

Спецификация формата DXF

DXF — это открытый формат файлов для обмена информацией между системами автоматизированного проектирования, который поддерживается почти всеми САД-системами.

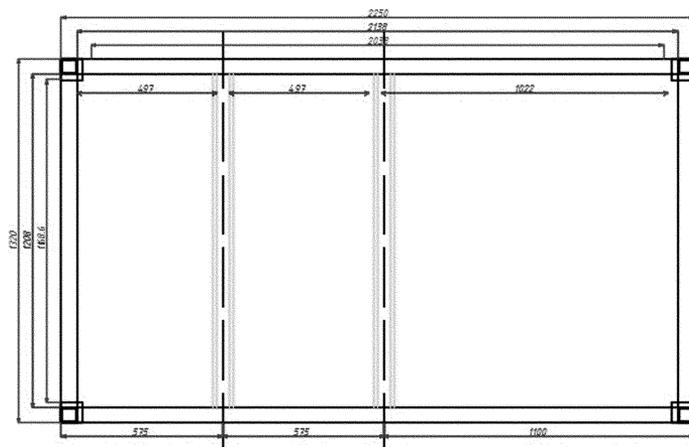


Рис. 2. Секция в сечении.

Сам формат представляет собой текстовый файл, заполненный группами — минимальными структурными единицами формата. Каждая группа состоит из кода группы и значения. Код группы определяет как тип значения (0–9 для строковых данных, 10–59 для вещественных чисел, 60–79 для целых чисел и т. д.), так и то, что содержится в значении группы. Так, код 0 используется для определения начала и конца секции, таблицы и так далее [3].

Структурно DXF-файл представляет собой набор секций. Каждая секция начинается с двух групп:

```
0
SECTION
2
SECTION_NAME
```

и заканчивается группой

```
0
ENDSEC
```

Порядок секций и их количество зависит от версии формата.

В качестве рабочей версии формата выбрана AC1009, что соответствует AutoCAD R11 или R12. Несмотря на то, что эта версия форма-

та является довольно старой, она поддерживает все требуемые примитивы для отрисовки секций и таблиц, но при этом менее требовательна к дополнительной информации, которая требуется AutoCAD для корректного отображения DXF файла.

Исходная библиотека в некоторых случаях генерирует некорректный DXF файл, а также содержит не все требуемые возможности для модуля отрисовки. Поэтому для использования библиотеки в составе модуля требовалась её доработка.

Так, написан класс ViewPort, который устанавливает область отображения, достаточную, чтобы при открытии файла в формате DXF чертёж попадал в область видимости без необходимости масштабирования.

Кроме того, AutoCAD использует свой шрифт (ISOCPEUR Italic) и если его не указать, то для текста по умолчанию будет использован шрифт Arial или другой системный шрифт.

Генерация чертежей

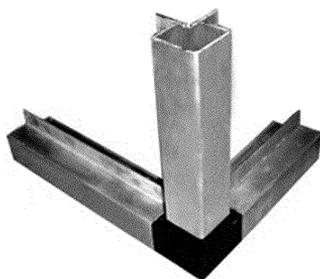


Рис. 3. Уголок и угловые профили.

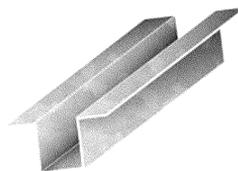


Рис. 4. Омега профиль.

Для генерации чертежей написан класс Drawer, на вход которого подаются геометрические характеристики объекта. Чертёж представляет собой изображение сечения в сечении и состоит из уголков, угловых профилей (см. рис. 3) и омега-профилей (см. рис. 4). Омега-профиль позволяет сократить расходы при сборке конструкции, он позволяет устанавливать панели непосредственно во внутрь профиля без крепления по периметру, без сварочных работ. Угловой профиль

представляет из себя угловую металлическую конструкцию, которая крепится снаружи к панелям.

По входным данным сначала отрисовывается сам каркас секции (см. рис. 5). Приведём код вывода внешней полилинии каркаса, внутренняя полилиния выводится аналогично, со смещением.

```
$this->document->append(new DxfLwPolyLine(
    array('points' => array(
        array($data->length, $data->height),
        array(0.0, $data->height),
        array(0.0, 0.0),
        array($data->length, 0.0),
        array($data->length, $data->height)
    )
    ,
    'width' => 0.15,
    'color' => 0)
));
```

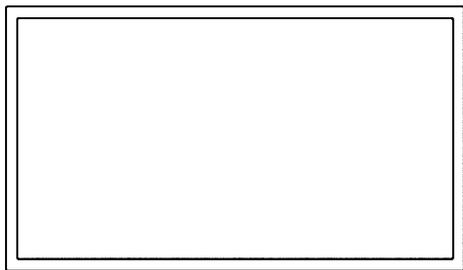


Рис. 5. Каркас секции.

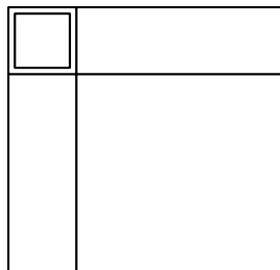


Рис. 6. Угловой профиль.

Затем генерируются уголки (см. рис. 6). Приведём пример кода для генерации внешнего контура уголка. Внутренний контур генерируется аналогично.

```
$this->document->append(new DxfLwPolyLine(
    array('points' => array(
        array($offset_x, $offset_y),
        array(0.0, $offset_y),
        array(0.0, 0.0),
```

```

    array($offset_x, 0.0),
    array($offset_x, $offset_y)
  ),
  'width' => 1.0,
  'color' => 0)
));

```

Аналогично генерируем уголки в остальных углах секции.

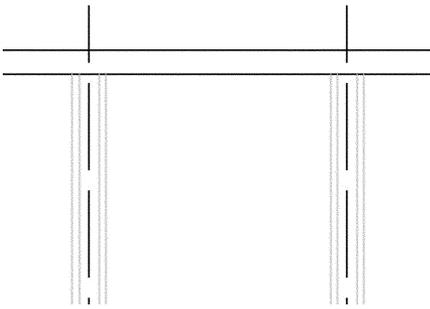


Рис. 7. Омега профили.

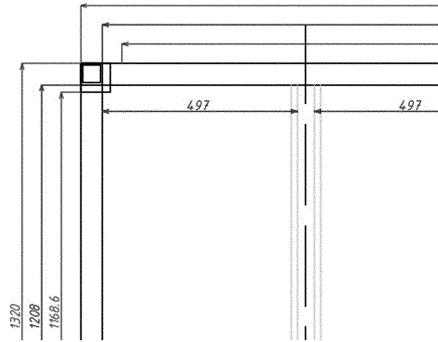


Рис. 8. Сноски с размерами.

После вывода уголков производится отрисовка омега-профилей (см. рис. 7). Приведём пример кода для отрисовки одной стороны омега-профиля. Вторая сторона отрисовывается аналогично.

```

$this->document->append(new DxfLine(
  array('points' => array(
    array($off2_x + $data->omegaPanels[$i]
      - $half_omega_profile_width - 15,
      0.0 + $offset_y),
    array($off2_x + $data->omegaPanels[$i]
      - $half_omega_profile_width - 15,
      $data->height - $offset_y)
  )
),
  'color' => 4
));

```

Далее выводятся сноски с размерами (см. рис. 8). Сначала выводятся две линии от стороны секции к которой выводится сноска, затем отрисовывается линия со стрелочками на обоих концах и над ней текст с размером стороны секции.

Отрисовка таблиц

Для генерации таблиц со спецификацией и характеристикой вентиляционного оборудования написан класс Document, который генерирует их согласно ГОСТу 21.602–79 — «Отопление, вентиляция и кондиционирование воздуха» [2]. Характеристика отопительно-вентиляционных систем выполняется в виде таблицы, которая содержит колонки с техническими характеристиками вентилятора, электродвигателя, воздухонагревателя, воздухоохладителя и фильтра (см. рис. 9). При отсутствии в системах отдельных видов оборудования соответствующие графы из таблицы исключают.

Вентилятор				Электродвигатель				Воздухонагреватель			
Тип	L, мЗ/ч	P, Па	n, об/мин	Тип	N, Вт	n, об/мин	фазность U, В	Тип	Т-ра нагрева, С		Расход тепла, кВт
									от	до	
4	5	6	7	8	9	10	11	12	13	14	15
ЭР50Ф-4, ДН. Д7. 1Р	3757	731	1149	-	1500	1500	Э- 400V 50Hz Y	HEATING	-28	18	46.29

Рис. 9. Таблица с характеристикой.

Для вывода характеристики вентиляционной установки в первую очередь генерируется шаблон страницы на которой в дальнейшем будет размещена таблица.

Сначала отрисовывается сам лист (формата А4).

```
$this->document->append(new DxfLine(
    array('points'=>array(
        array($base_x+$this->width, $base_y+$this->height),
```

```

    array($base_x+0.0, $base_y+$this->height)
  ),
  'color' => 0
))
);
// ...

```

Далее выводится штамп согласно шаблону чертежа (см. рис. 10). Приведём код отрисовки первой линии штампа.

```

$this->document->append(new DxfLine(
  array('points'=>array(
    array($base_x+$this->width - 500,
      $base_y+$this->height - 500),
    array($base_x+2000.0, $base_y+$this->height - 500)
  )),
  'color' => 0
))
);
// ...

```

Затем выводится текст штампа. Приведём код вывода первой строки.

```

$this->document->append(new DxfText(
  array('text' => "Согласовано",
    'point' => array($base_x+400, $base_y+10500),
    'rotation' => 90,
    'height' => 360)));
// ...

```

			<i>Согласовано</i>			
<i>Инв. № подл.</i>	<i>Подпись и дата</i>	<i>Взам. инв. №</i>				

Рис. 10. Штамп.

Затем шаблон для информация о проекте. Он отличается в зависимости от того, первая это страница или нет.

Здесь также сначала генерируется таблица для данных о проекте (см. рис. 11). Приведём код для вывода первой линии шаблона.

```
$this->document->append(new DxfLine(
array('points'=>array(
array($base_x + $this->width - 500, $base_y + 500),
array($base_x + 23000.0, $base_y + 500)
),
'color' => 0)));
// ...
```

А затем уже выводятся текстовые данные в этой таблице. Приведём код вывода первой строки.

```
$this->document->append(new DxfText(
array('text' => "Изм.",
'point' => array($base_x + 23100, $base_y + 3600),
'height' => 260)));
// ...
```

					<i>Д015-00001-01</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>Вентиляция и кондиционирование</i>	<i>Стадия</i>	<i>Лист</i>	<i>Листов</i>
<i>Разраб.</i>						А	1	2
<i>Проверил</i>					<i>Характеристика вентиляционного оборудования</i>			
<i>Утвердил</i>								



						<i>Д015-00001-01</i>		<i>Лист</i>
								2
<i>Изм.</i>	<i>Кол уч.</i>	<i>Лист</i>	<i>№ док.</i>	<i>Подпись</i>	<i>Дата</i>			

Рис. 11. Информация о проекте.

После отрисовки шаблона генерируется сама таблица с характеристикой. По большей части генерация таблицы представляет собой набор одинаковых линий, расположенных с определённым смещением друг от друга. Приведём код отрисовывающий одну из линий.

```

$this->document->append(new DxfLine(
    array('points'=>array(
        array($base_x+$this->width - 1000,
            $base_y+$this->height - 1000),
        array($base_x+2500.0, $base_y+$this->height - 1000)
    ),
    'color' => 0
)));
// ...

```

Далее, выводятся подписи в колонках этой таблицы (см. рис. 12). Приведём код вывода первой подписи.

```

$this->document->append(new DxfText(
    array('text' => "Обозначение",
        'point' => array($base_x+2500,
            $base_y+$this->height - 1000 - 1630),
        'height' => 200)));

```

Далее выводим строки таблицы в которых будут размещены данные.

```

$count = ($this->currentPage==0?23:31);
for ($i=0;$i<$count;$i++) {
$this->document->append(new DxfLine(
    array('points'=>array(
        array($base_x+2500, $base_y+$bottom + ($i*840)),
        array($base_x+41000, $base_y+$bottom + ($i*840))
    ),
    'color' => 0)));
}

```

После генерации таблицы, выводятся сами данные.

Аналогичным образом производится отрисовка спецификации вентиляционного оборудования.

Генерация трёхмерной установки

3D установка формируется набором примитивов. Для рендеринга OpenJsCad использует специальный формат jscad, который содержит

непосредственно сам код отрисовки [1]. Этот код выполняется в браузере и результат его работы рендерится средствами WebGL.

В целях рендеринга установки написано несколько функций для формирования каркаса и секций вентиляционной установки на JavaScript.

<i>Позиция</i>	<i>Наименование и техническая характеристика</i>	<i>Тип, марка обозначение документа, опросного листа</i>		<i>Единица Изменения</i>	<i>Количество</i>	<i>Масса единицы, кг.</i>	<i>Примечание</i>
<i>1</i>	<i>2</i>	<i>3</i>		<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

Рис. 12. Колонки таблицы.

Каркас секции формируется из угловых профилей и уголков. Угловой профиль для каждой стороны каркаса формируется парой параллелограммов.

После чего во всех углах полученного каркаса выставляются уголки, которые также формируются из параллелограммов. Далее вся конструкция объединяется в единый блок для простоты манипулирования им.

```
function framework(w,h,l,scale) {
var lineW = cube({size: [scale*5,scale*50,w]})
                .union(cube({size: [scale*50,scale*5,w]}));
var fw_cube = cube({size: [scale*50,scale*50,scale*50]});
```

Для отрисовки каркаса секции копируются одинаковые по размеру угловые профили, разворачиваются и перемещаются на положенное место.

Приведём пример кода для формирования одной из сторон каркаса секции.

```
var line1 = lineH;
var line2 = lineH.rotateZ(90).translate([1,0,0]);
var line3 = lineH.rotateZ(-90).translate([0,w,0]);
var line4 = lineH.rotateZ(-180).translate([1,w,0]);
```

Уголки секции формируются аналогично каркасу: на основе одного уголка формируются остальные путём перемещения и разворота.

Приведём пример кода для формирования уголков на одной из сторон каркаса.

```
var cube1 = fw_cube.translate([0,0,0]);
var cube2 = fw_cube.rotateZ(90).translate([1,0,0]);
var cube3 = fw_cube.rotateZ(180).translate([1,w,0]);
var cube4 = fw_cube.rotateZ(-90).translate([0,w,0]);
```

Далее, мы собираем каркас в единый объект используя операцию объединения (union).

Секции также формируются из примитивов. Например, вентилятор формируется из параллелограмма и цилиндра. Ещё один параллелограмм и цилиндр являются вспомогательными и вырезаются из основных фигур для создания отверстий.

```
var c1 = cube({size:[w*0.3,w/4,w/4]});
var c2 = cylinder({r:w/4,h:w/4});
var c = c1.union(c2);
```

Далее, для формирования установки целиком написан класс JsCadDrawer, который позволяет генерировать jscad файл с необходимыми блоками для формирования вентиляционной установки.

В нём были реализованы методы для добавления блоков в приток и вытяжку вентиляционной установки, а также метод генерации из этих блоков jscad файла установки.

Приведём пример кода, который добавляет секцию в приток. Сначала мы высчитываем смещение для блока. Затем проверяем какого типа секция, если рекуператор, то отрисовываем внизу, если нет, то в зависимости от того, встречали мы уже рекуператор или нет приток будет отрисовываться или сверху или снизу.

```
function appendSupplyBlock($block){
  $block->offset_x=$this->offset_supply;
  $isRecuperator = is_a($block, 'RecuperatorSection');
  $block->offset_y=($this->ti_recuperator?0
    :((($isRecuperator)?0:$this->height));
  $this->offset_supply += $block->width;
  if ($isRecuperator) {
    $this->ti_recuperator = true;
```

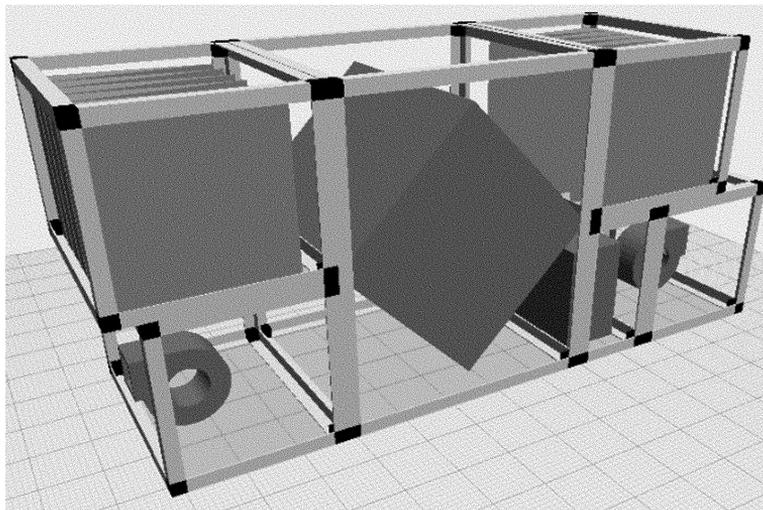


Рис. 13. Отрисованная 3D-установка.

```
} else {  
    $this->height = $block->height;  
}  
$this->blocks[] = $block;  
}
```

Добавление секций в вытяжку происходит аналогичным образом. Вокруг класса JsCadDrawer написана обёртка, которая по исходным данным формирует блоки и добавляет в приток или вытяжку установки.

Приведём пример кода для формирования установки. Сначала мы создаём экземпляр отрисовщика и передаём в его конструктор различные параметры. Затем пробегаемся по блокам, которые входят в приток установки и в зависимости от типа блока создаём экземпляр соответствующего класса. Аналогично добавляются блоки в вытяжку установки. В результате на выходе отрисовщика мы получаем файл с 3D-изображением установки.

```
$d = new JsCadDrawer($total_width_supply, ...);  
foreach ($supply_blocks as $b) {  
    $blk = null;
```

```
switch($b["type"]) {  
  case "v":  
    $blk = new VentilatorSection($b["w"],$b["h"],$b["l"]);  
    break;  
  // ... Другие секции ...  
}  
if ($blk!=null) {  
  $d->appendSupplyBlock($blk);  
}
```

Заключение

В результате выполненной работы реализован модуль генерации спецификации, характеристики, а также чертежей в формате DXF. Кроме того, реализован модуль трёхмерной отрисовки установки для отображение в веб-браузере.

Список литературы

- [1] <https://github.com/Spiritdude/OpenJSCAD.org/wiki/User-Guide>
- [2] http://www.apic.ru/standards/pdf/standards/GOST21_602_79.pdf
- [3] http://images.autodesk.com/adsk/files/acad_dxf0.pdf