

コンテスト運営を支援する自動集計システム



2023年3月22日

無線部開発班 JG1VPP

nextzlog.dev

目次

第 1 章	はじめに	3
1.1	経緯	3
1.2	特色	3
第 2 章	従来方式	4
第 3 章	書類提出	5
3.1	書類提出の開始	5
3.2	交信記録の添付	6
3.3	提出書類の確認	7
第 4 章	起動方法	8

第1章 はじめに

本稿で解説する**自動集計システム**は、アマチュア無線のコンテストの効率的な運営を支援する、**ウェブシステム**である。ALLJA1 コンテストを対象に、参加者の集計と結果発表を迅速化する目的で整備された。2014 年以来の運用実績がある。

1.1 経緯

ALLJA1 コンテストは毎年 6 月の開催だが、2009 年に部員が数名に減勢した我が無線部では、開催困難な状況に陥った。運営業務は、以下の 4 段階に区分できるが、書類審査の負担が重く、恒常的に結果発表が年度末まで遅れる状況だった。

開催前の業務	規約策定と告知
開催中の業務	開催状況の把握
審査中の業務	書類受付・書類審査
審査後の業務	結果発表・賞状発送

2010 年の増勢により、当面は開催を継続する方針に決着したが、外部に運営を委託する可能性も検討される状況だった。駒場には委託に抵抗を感じる学生もあり、単独での運営を継続するために整備を始めたのが、下記のシステム群である。

ATS-1 型	2012 年	第 25 回	… 部分的なサマリーシートの自動処理の実現
ATS-2 型	2013 年	第 26 回	… 書類解析の厳密かとウェブ書類受付の実現
ATS-3 型	2014 年	第 27 回	… 書類解析と暫定結果発表のリアルタイム化
ATS-4 型	2017 年	第 30 回	… 自動集計システムとコンテスト規約の分離

2013 年には、交信記録を完全に自動処理できる ATS-2 型を試作し、悲願だった、締切から 2 日での結果速報を達成した。2021 年には、従来の ALLJA1 コンテストに加え、JS2FVO らの発案でリアルタイムコンテストの運営業務にも対応した。

1.2 特色

ATS-4 型の特色は、任意のコンテストを**ドメイン特化言語**で定義でき、規約の変更や移植に柔軟に対応可能な点にある。ATS-4 型の移植や設置の相談は、nextzlog/todo で受け付ける。以下に、既に対応を果たした規約と、その実装を示す。

電通大コンテスト	https://github.com/nextzlog/ats4/blob/master/conf/rules/JA1ZGP/uec.rb
多摩川コンテスト	https://github.com/nextzlog/ats4/blob/master/conf/rules/JI1YEG/tama.rb

規約は、Ruby で記述される。以下に、簡単な例を示す。得点計算が特殊な場合は、**クラス**を拡張する形で、対応できる。Ruby 以外では、ALLJA1 コンテストなど、交信記録の解析方法や得点計算が複雑な場合に、LISP を併用する例がある。

```
RULE = PlainProgram.new('QSO PARTY')

RULE.add(PlainSection.new('14MHz CW', [Band.new(14000)], [Mode.new('CW')]))
RULE.add(PlainSection.new('21MHz CW', [Band.new(21000)], [Mode.new('CW')]))
RULE.add(PlainSection.new('28MHz CW', [Band.new(28000)], [Mode.new('CW')]))
RULE.add(PlainSection.new('50MHz CW', [Band.new(50000)], [Mode.new('CW')]))

RULE.add(PlainSection.new('14MHz PH', [Band.new(14000)], [Mode.new('SSB'), Mode.new('FM')]))
RULE.add(PlainSection.new('21MHz PH', [Band.new(21000)], [Mode.new('SSB'), Mode.new('FM')]))
RULE.add(PlainSection.new('28MHz PH', [Band.new(28000)], [Mode.new('SSB'), Mode.new('FM')]))
RULE.add(PlainSection.new('50MHz PH', [Band.new(50000)], [Mode.new('SSB'), Mode.new('FM')]))
```

第2章 従来方式

我が無線部では、開催後の書類受付の要領を抜本的に見直し、書類の曖昧性を排除して、自動処理する方法を模索した。日本国内のコンテストでは、JARL が推奨する **サマリーシート** を、電子メールに添付して提出する方法が標準的である。

```
<SUMMARYSHEET VERSION=R2.0>
<CALLSIGN>JA1ZLO</CALLSIGN>
<TOTALSCORE>64</TOTALSCORE>
<CONTESTNAME>ALLJA1</CONTESTNAME>
<CATEGORYCODE>XMAH</CATEGORYCODE>
<LOGSHEET TYPE=ZLOG>
mon day time callsign      sent      rcvd      multi    MHz mode pts memo
  6   1 0932 JA1YAD       100110     59100110   100110    14 SSB  1
  6   1 0956 JA1YYE       100110     5913009    13009    28 SSB  1
  6   1 1002 JA1YXP       100110     59134404   134404    50 AM   1
  6   1 1027 JR1ZTT       100110     591420     1420     21 SSB  1
  6   1 1629 JA1YCG       100110     59110109   110109    7 SSB  1
  6   1 1637 JA1YDU       100110     5991216    1216      7 CW   1
  6   1 1717 JA1ZGP       100110     5991009    1009      7 CW   1
  6   1 1738 JA1YGX       100110     59100105   100105    7 SSB  1
</LOGSHEET>
</SUMMARYSHEET>
```

交信記録に加え、参加者の氏名や連絡先に、参加部門を記載する。しかし、曖昧性が高く、自動処理には不適切である。例えば、ATS-1 型の開発段階では、参加部門を確定する際に、その曖昧さ故に、稚拙な判別方法を採用する必要があった。

電信と電話の判別 要素 CATEGORYNAME の値に語「電話」があれば**電信電話部門**
運用エリアの検査 要素 CATEGORYNAME の値に語「内」があれば**関東エリア部門**
社団と個人の判別 要素 CATEGORYNAME の値に語「マルチ」があれば**社団局部門**

また、交信の日時や相手や周波数を記載した LOGSHEET の部分には、規格化された書式がなく、実質的には自由欄だった。交信を記録するソフトウェア毎に独自の書式が乱立して、構造や属性の形式的な定義も提供されず、曖昧な状態である。

```
<LOGSHEET TYPE=JA1ZLO-ORIGINAL-FORMAT>
```

属性には、複数の解釈の余地があり、以下の2行は、規約次第で、同じ意味になる場合も、異なる意味になる場合もある。

```
2015-06-07 09:01 JA1YWX 100105
2015-06-07 09:01 JA1YWX 59100105
```

国際的なコンテストの場合は、交信記録の書式を厳格に規定した事例があり、Cabrillo や ADIF が代表的な書式である。前者はコンテスト毎に詳細が異なり、交信を記録するソフトウェア側で個別のコンテストの書式に対応する必要がある。

```
START-OF-LOG: 3.0
CALLSIGN: JA1ZLO
QS0: 7000 CW 1919-08-10 0364 JA1ZLO 599 114514 JA1YWX 599 889464 0
QS0: 7000 CW 1919-08-10 0364 JA1ZLO 599 114514 JA1YWX 599 889464 0
```

後者は、規格が厳密で拡張性もあるが、名前空間の概念がなく、独自に定義された属性の名前が重複する可能性がある。独自定義の属性を自動的に検証する仕組みも、参加部門の曖昧さを解決する仕組みもなく、運用次第では曖昧さが残る。

```
<CALL:6>QI1JDS<QS0_DATE:8>20170604<time_on:6>000000<MODE:2>CW<band:3>10m<RST_RCVD:3>599<SRX:4>1005<eor>
<CALL:6>QD2LVE<QS0_DATE:8>20170604<time_on:6>000100<MODE:2>CW<band:3>20m<RST_RCVD:3>599<SRX:4>1336<eor>
```

第3章 書類提出

第2章で提起した問題意識から、我が無線部ではウェブ提出の仕組みを構築して、電子メールでの書類受付を廃止した。ATS-3型の開発では、PCの操作が苦手な参加者に配慮して、無駄な画面遷移を排除し、**ユーザビリティ**の確保に努めた。

3.1 書類提出の開始

参加者は交信記録を準備して、ATS-4型にアクセスする。書類提出のボタンを押すと、Fig. 3.1に示す画面が表示される。呼出符号や連絡先を記入し、運用場所と部門を選ぶ。第3.2節に解説する手順で交信記録を添付し、提出のボタンを押す。

書類提出

提出内容をご確認ください

⚠ 未入力の項目があります

参加局の情報

呼出符号	移動運用の例: JA1ZLO/1	⚠
運用場所	東京都	▼
参加部門	1エリア内 社団 電信限定 オールバンド部門	▼
お名前	我が無線部	
ご住所	東京都目黒区駒場	

Fig. 3.1: entry sheet.

必要な情報が空欄の場合は、赤字で表示されるので、修正して提出する。この仕組みにより、書類の曖昧さが排除できる。なお、JARLが推奨するサマリーシートとは異なり、宣誓欄や資格や署名などの記入欄を削除して、画面を簡素化した。

3.2 交信記録の添付

第 3.1 節で必要な情報を記入した後で、Fig. 3.2 に示すファイル選択画面で、交信記録を添付して、提出のボタンを押す。これで、種類提出は完了である。登録内容を確認する画面が表示され、誤りがあれば、締切までに何度でも再提出できる。

交信記録

JARL サマリーシートをアップロードした場合は、交信記録の部分のみを読み取ります

ats3test.zlo

選択

対応済みの電子ログのフォーマットの一覧を見る

Fig. 3.2: upload form for the operational log.

ATS-4 型は、交信記録の書式を自動的に判別する機能を備える。以下の書式に対応済みで、殆どの交信記録を網羅する。自動判別の精度の都合で、CTESTWIN の場合は LG8 を、zLog の場合は ZLO を、それ以外の場合は ADIF を推奨する。

- 形式言語型 qxml, ADIF
- バイナリ型 CTESTWIN(LG8), zLog(ZLO)
- テキスト型 CTESTWIN(TXT), zLog(TXT), zLog(ALL), Cabrillo(CQWW), JARL R2.0

参加者の便宜を図るため、JARL が推奨するサマリーシートの提出にも対応したが、LOGSHEET 以外の内容は無視される。また、確実に読み取れる保証がなく非推奨だが、各種のソフトウェアが出力するプレーンテキストの書式にも対応した。

(zLog DOS)											y	年		
MM	dd	HHmm	CCCCCCCC	SSSSSSSSSS	RRRRRRRRRR	*****	BBBB	EEEE	***	NNNN	M	月		
6	4	0117	QV1DOK	599100110	599120103		14	CW	1		d	日		
6	4	0151	QC2SOA	599100110	5991308		50	CW	1		H	時		
											m	分		
(zLog ALL)											C	呼出符号		
yyyy/MM/dd	HH:mm	CCCCCCCC	sss	SSSSSS	rrr	RRRRRR	*****	*****	BBBB	EEEE	**	NNNN	B	周波数帯
2017/06/04	01:17	QV1DOK		599 100110	599 120103	120103-		14	CW	1			F	周波数
2017/06/04	01:51	QC2SOA		599 100110	599 1308	1308 -		50	CW	1			E	変調方式
											S	送信符号		
											R	受信符号		
(CTESTWIN)											s	送信 RST		
****	MM/dd	HHmm	CCCCCCCC	BBBBBB	EEEE	SSSSSSSSSS	RRRRRRRRRR					r	受信 RST	
1	6/ 4	0117	QV1DOK	14MHz	CW	599100110	599120103					O	運用者名	
2	6/ 4	0151	QC2SOA	50MHz	CW	599100110	5991308					N	備考	
											*	無視		

以上のプレーンテキストの書式は、固定長の書式か、備考欄のみ可変長の書式と見做しベストエフォートで処理される。なお、JARL が 2016 年に改訂したサマリーシート R2.0 の LOGSHEET の部分は、可変長とする。+は任意長の反復を表す。

(jarl)
yyyy-MM-dd HH:mm B+ E+ C+ s+ S+ r+ R+

3.3 提出書類の確認

書類提出が終わると、自動集計システムは、内容を簡単に検査して暫定の得点を計算し、参加者に確認画面を送り返す。

書類提出の完了

下記の内容で提出書類を受理しました。訂正が必要な場合は期間内に限り再提出できます。

参加局の情報

1エリア内 社団 電信限定 オールバンド部門

受付時刻	呼出符号	運用地	お名前	メール	ご住所
2017.02.26.15.11.35	JA1ZLO	東京都	我が無線部 様	allja1@ja1zlo.u-tokyo.org	東京都目黒区駒場

(a) general profile.

暫定の得点

交信局数8 & 獲得マルチ数8 = **64点**

(b) temporary score.

有効な交信

相手局の呼出符号や運用地が正しく処理されているかご確認ください

	交信日時	相手局	周波数帯	通信方式	運用地
001	2014年6月1日 09時35分	JA1YWX	21MHz	CW	100105
002	2014年6月1日 11時06分	JA7YCQ	14MHz	CW	06
003	2014年6月1日 16時02分	JA1ZLO	1.9MHz	CW	100110
004	2014年6月1日 16時07分	JA1YGX	3.5MHz	CW	100105
005	2014年6月1日 16時37分	JA1YDU	7MHz	CW	1216
006	2014年6月1日 17時17分	JA1ZGP	7MHz	CW	1009

(c) operational log.

Fig. 3.3: submission certificate.

参加者は、交信記録が適切に処理された旨を自分で確認する必要がある。必要なら、締切までに修正して再提出できる。

第4章 起動方法

Windows や UNIX 系 OS で Docker を導入し、bash で以下のコマンドを実行すると、ATS-4 型が localhost で起動する。

```
cat << EOS > docker-compose.yaml
version: '3'
services:
  ATS4:
    image: ghcr.io/nextzlog/ats4:master
    ports:
      - 9000:9000
    volumes:
      - ./ats/data:/ats/data
      - ./ats/logs:/ats/logs
      - ./ats.conf:/ats/conf/ats.conf
      - ./rules.rb:/ats/conf/rules.rb
    command: /ats/bin/ats4
  www:
    image: nginx:latest
    ports:
      - 80:80
    volumes:
      - ./proxy.conf:/etc/nginx/conf.d/default.conf
EOS

echo -n 'enter mail hostname: '; read host
echo -n 'enter mail username: '; read user
echo -n 'enter mail password: '; read pass
echo -n 'enter server domain: '; read name

cat << EOS > ats.conf
play.mailer.host=$host
play.mailer.port=465
play.mailer.ssl=true
play.mailer.user="$user"
play.mailer.password="$pass"
play.mailer.mock=false
ats4.rules=/rules.rb
EOS

cat << EOS > rules.rb
require 'rules/sample/plain'
RULE
EOS

cat << EOS > proxy.conf
server {
  server_name $name;
  location / {
    proxy_pass http://ATS4:9000;
    location ~ /admin {
      allow 127.0.0.1;
      deny all;
    }
  }
}
EOS

docker compose up -d
```