Middleware for Best-Effort Third-Party Monitoring

No Author Given

No Institute Given

1 Monitoring Specifications

There is a basic type system for event streams:

$$\text{Event Data Type } \tau^E \supseteq \texttt{int} \mid \texttt{bool} \mid \texttt{byte} \mid \texttt{string} \mid \tau^E[] \mid \tau \times \tau \mid \tau + \tau$$

We assume a language with the following constructs:

$$\begin{aligned}
\mathit{VAR} &\supseteq x \\
\mathit{TYPE} &\supseteq \tau^E \\
\mathit{EXPR} &\supseteq \mathit{VAR} \mid n \mid \texttt{true} \mid \texttt{false} \mid \mathit{EXPR} + \mathit{EXPR} \mid \mathit{EXPR} \wedge \mathit{EXPR} \ldots \\
\mathit{CODE} &\supseteq \{\overline{\mathit{CODE}}\} \mid \mathit{VAR} \ : \ \mathit{TYPE} \ = \ \mathit{EXPR}; \mid \mathit{VAR} \ = \ \mathit{EXPR}; \mid \mathit{EXPR}; \mid \texttt{if}(\mathit{EXPR}) \ \mathit{CODE} \, \texttt{else} \ \mathit{CODE}
\end{aligned}$$

$$\begin{aligned}
\mathit{FIELDDECL} &::= \mathit{fieldname} : \mathit{TYPE} \\
\mathit{EVDEF} &::= \mathit{eventname}(\overline{\mathit{FIELDDECL}}) \\
\mathit{STYPEDEF} &::= \texttt{stream type } \mathit{stypename} \ \{(\overline{\mathit{FIELDDECL}})\} \ \{\texttt{extends } \mathit{stypename} \ \{(\overline{\mathit{EXPR}})\}\} \\
&\qquad\quad '\{' \ \overline{\mathit{EVDEF} \ \{\texttt{creates } \mathit{stypename}\}} \ '\}'
\end{aligned}$$

Fig. 1. Event Streams

$$\begin{aligned}
\mathit{ACTION} &::= \mathtt{drop} \mid \mathtt{forward} \mid \mathtt{forward}\ \mathit{eventname}(\overline{\mathit{EXPR}})\\
\mathit{CASE} &::= \mathit{ACTION} \mid \mathtt{if}(\mathit{EXPR})\ \mathtt{then}\ \mathit{CASE}\ \mathtt{else}\ \mathit{CASE}\\
\mathit{PROCESSORRULE} &::= \mathtt{on}\ \mathit{eventname}\ \{(\overline{\mathit{VAR}})\}\ \mathit{CASE}\\
&\quad\mid\ \mathtt{on}\ \mathit{eventname}\ \{(\overline{\mathit{VAR}})\}\ \mathtt{creates}\ \{\mathtt{at\ most}\ n\}\ \mathit{stypename}\\
&\qquad\quad \{\mathtt{process\ using}\ \mathit{processorname}\ \{(\overline{\mathit{EXPR}})\}\}\\
&\qquad\quad \mathtt{to}\ \mathit{BUFFERKIND}\ \mathtt{include\ in}\ \mathit{bufgroupname}\ \mathit{CASE}\\
\mathit{PROCESSORDEF} &::= \mathtt{stream\ processor}\ \mathit{processorname}\ \{(\overline{\mathit{VAR}})\}\\
&\qquad\quad :\ \mathit{stypename}\ \{(\overline{\mathit{EXPR}})\}\ \rightarrow\ \mathit{stypename}\ \{(\overline{\mathit{EXPR}})\}\\
&\qquad\quad \{\mathtt{extends}\ \mathit{processorname}\ \{(\overline{\mathit{EXPR}})\}\}\ '\{'\ \overline{\mathit{PROCESSORRULE}}\ '\}'
\end{aligned}$$

Fig. 2. Stream Processors

$$\begin{aligned}
\mathit{BUFFERKIND} &::= \mathtt{autodrop}(n) \mid \mathtt{infinite} \mid \mathtt{blocking}(n)\\
\mathit{ESRCDEF} &::= \{\mathtt{dynamic}\}\ \mathtt{event\ source}\ \mathit{evsrcname}\ \{(\overline{\mathit{FIELDDECL}})\}\ \{[n]\}\ :\ \mathit{stypename}\\
&\qquad \{\mathtt{process\ using}\ \mathit{processorname}\ \{(\overline{\mathit{EXPR}})\}\}\\
&\qquad \mathtt{to}\ \mathit{BUFFERKIND}\ \{\mathtt{include\ in}\ \mathit{bufgroupname}\}
\end{aligned}$$

Fig. 3. Performance Layer Specification

$$\begin{aligned}
\mathit{norall} &::= n \mid \mathtt{all}\\
\mathit{ORDEREXP} &::= \mathtt{round\ robin} \mid \mathtt{\$}\mathit{fieldname}\\
\mathit{BUFGROUPDEF} &::= \mathtt{buffer\ group}\ \mathit{bufgroupname} : \mathit{stypename}\ \{\mathtt{order\ by}\ \mathit{ORDEREXP}\}\\
&\qquad \{\mathtt{includes}\ \overline{\mathit{evsrcname}\ \{[\mathit{norall}]}\}\}\\
\mathit{EVSRCREF} &::= \mathit{evsrcname}\ \{[n]\}\\
\mathit{varorevsrcname} &::= \mathit{VAR} \mid \mathit{EVSRCREF}\\
\mathit{MATCHFUNDEF} &::= \mathtt{match\ fun}\ \mathit{matchfunname}[\overline{\mathit{varorevsrcname}}](\overline{\mathit{VAR}}) = \overline{\mathit{BUFFERMATCHEXP}}
\end{aligned}$$

Fig. 4. Advanced Features

$$
\begin{array}{rcl}
\mathit{varorinteger} & ::= & \mathit{VAR} \mid n \\
\mathit{BUFFERMATCHEXP} & ::= & \mathit{matchfunname}[\overline{\mathit{varorevsrcname}}](\overline{\mathit{varorinteger}}) \mid \mathit{EVSRCREF}\ :\ \texttt{done} \\
 & \mid & \mathit{EVSRCREF}\ :\ \texttt{nothing} \mid \mathit{EVSRCREF}\ :\ \overline{\mathit{eventname}(\overline{\mathit{VAR}})}\ '|'\ \overline{\mathit{eventname}(\overline{\mathit{VAR}})} \\
 & \mid & \texttt{choose}\ \{\texttt{first/last}\ \{n\}\}\ \overline{\mathit{evsrcname}}\ \texttt{from}\ \mathit{bufgroupname} \\
\mathit{ARBITERRULESTMT} & ::= & \text{like } \mathit{CCODE}\text{, without } \texttt{return}\text{, but with} \\
 & \mid & \texttt{\$drop}\ n\ \texttt{from}\ \mathit{EVSRCREF} \mid \texttt{\$yield}\ \mathit{eventname}(\overline{\mathit{EXPR}}) \\
 & \mid & \texttt{\$switch to}\ \mathit{rulesetname} \mid \texttt{\$add}\ \mathit{EVSRCREF}\ \texttt{to}\ \mathit{bufgroupname} \\
 & \mid & \texttt{\$remove}\ \mathit{EVSRCREF}\ \texttt{from}\ \mathit{bufgroupname} \\
 & & \text{and access to fields in event sources in expressions using } \$\mathit{evsrcname.fieldname} \\
\mathit{ARBITERRULEDEF} & ::= & \texttt{on}\ \overline{\mathit{BUFFERMATCHEXP}}\ \texttt{where}\ \mathit{EXPR}\ '\{'\ \overline{\mathit{ARBITERRULESTMT}}\ '\}' \\
 & \mid & \texttt{choose}\ \{\texttt{first/last}\ \{n\}\}\ \overline{\mathit{evsrcname}}\ \texttt{from}\ \mathit{bufgroupname}\ \texttt{where}\ \mathit{EXPR} \\
 & & \quad '\{'\ \overline{\mathit{ARBITERRULEDEF}}\ '\}' \\
\mathit{RULESETDEF} & ::= & \texttt{rule set}\ \mathit{rulesetname}\ '\{'\ \overline{\mathit{ARBITERRULEDEF}}\ '\}' \\
\mathit{ARBITERDEF} & ::= & \texttt{arbiter} : \mathit{stypename}'\{'\ \overline{\mathit{RULESETDEF}}\ '\}'
\end{array}
$$

Fig. 5. Arbiter Specification

$$
\begin{array}{rcl}
\mathit{MONITORRULEDEF} & ::= & \texttt{on}\ \mathit{eventname}(\overline{\mathit{VAR}})\ \{\texttt{where}\ \mathit{EXPR}\}\ \{\overline{\mathit{CODE}}\} \\
\mathit{MONITORDEF} & ::= & \texttt{monitor}\{\overline{\mathit{MONITORRULEDEF}}\}
\end{array}
$$

Fig. 6. Monitor Specification

$$
\begin{array}{rcl}
\mathit{COMPONENT} & ::= & \mathit{STYPEDEF} \mid \mathit{PROCESSORDEF} \mid \mathit{ESRCDEF} \mid \mathit{BUFGROUPDEF} \mid \mathit{MATCHFUNDEF} \\
 & \mid & \mathit{GLOBALDEF} \mid \mathit{STARTUPDEF} \mid \mathit{CLEANUPDEF} \\
\mathit{GLOBALDEF} & ::= & \texttt{globals}\ '\{'\ \mathit{CODE}\ '\}' \\
\mathit{STARTUPDEF} & ::= & \texttt{startup}\ '\{'\ \mathit{CODE}\ '\}' \\
\mathit{CLEANUPDEF} & ::= & \texttt{cleanup}\ '\{'\ \mathit{CODE}\ '\}' \\
\mathit{PROGRAM} & ::= & \overline{\mathit{COMPONENT}}\ \mathit{ARBITERDEF}\ \mathit{MONITORDEF}
\end{array}
$$

Fig. 7. Full Programs